

*Donatie din partea autorilor pentru
Biblioteca Centrală Universitară
4 iulie 2003 Th. Hristea
H. Hristea*

Building Awareness in Language Technology

Papers of the Romanian Regional Information Centre
for Human Language Technology

81/390

Edited by

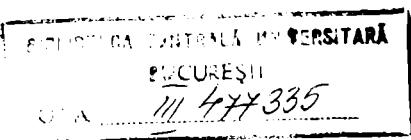
Florentina Hristea
University of Bucharest

Marius Popescu
University of Bucharest

10043011
7.7

Editura Universității din București

2003



D170/04

Tiparul s-a executat sub c-da nr. 1038/2003 la
Tipografia Editurii Universității din București

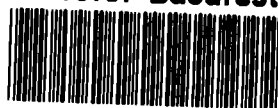
© Editura Universității din București

Șos. Panduri, 90-92, București - 76235; Telefon/Fax: 410.23.84

B.C.U. București

E-mail: editura@unibuc.ro

Internet: www.editura.unibuc.ro



C20040811

Descrierea CIP a Bibliotecii Naționale a României
Building Awareness in Language Technology: Papers of
the Romanian Regional Information Centre for
Human Language Technology / edited by Florentina
Hristea, Marius Popescu – București: Editura
Universității din București, 2003
Bibliografie
ISBN 973-575-748-6

I. Hristea, Florentina (ed.)

II. Popescu, Marius (ed.)

81 •

TABLE OF CONTENTS

<i>Editors' Forward</i>	1
Galia Angelova <i>Objectives of BALRIC-LING</i>	3
I. GRAMMATICAL FORMALISMS AND THEIR USAGE IN THE CASE OF THE ROMANIAN LANGUAGE. CORRESPONDING TOOLS FOR CORPORA ANNOTATION	
Florentina Hristea and Marius Popescu <i>A Dependency Grammar Approach to Syntactic Analysis with Special Reference to Romanian</i>	9
Marius Popescu <i>Dependency Grammar Annotator</i>	17
Emil Ionescu <i>Head-Driven Phrase Structure Grammar – A General Presentation</i>	35
The RORIC-LING Bulletin <i>months 1-6</i>	79
II. SEMIAUTOMATIC GENERATION OF WORDNET TYPE ROMANIAN SYNSETS AND CLUSTERS	
Florentina Hristea <i>On the Semiautomatic Generation of WordNet Type Synsets and Clusters with Special Reference to Romanian</i>	113
Theodor Hristea <i>Some Linguistic Comments Concerning the Obtained Output</i>	153
The RORIC-LING Bulletin <i>months 7-12</i>	159

**III. A THEORETICAL SPECIFICATION CONCERNING A
MORPHOLOGICAL MODEL FOR ROMANIAN**

Theodor Hristea and Cristian Moroianu
Generation of Romanian Noun and Adjective Forms 185

Emil Ionescu
Premises of a Morphological Dictionary of Romanian 203

The RORIC-LING Bulletin
months 13-18 211

CUPRINS

<i>Cuvântul editorilor</i>	223
Galia Angelova <i>Obiectivele proiectului BALRIC-LING</i>	227
I. FORMALISME GRAMATICALE ȘI UTILIZAREA LOR ÎN CAZUL LIMBII ROMÂNE. INSTRUMENTE CORESPUNZĂTOARE PENTRU ADNOTAREA CORPUSURILOR	
Florentina Hristea și Marius Popescu <i>Gramatici de dependență și gramatici WG</i>	233
Marius Popescu <i>Dependency Grammar Annotator</i>	247
Ana-Maria Barbu și Emil Ionescu <i>Teorii gramaticale contemporane: Gramatica centrilor de sintagmă (HPSG)</i>	265
Buletinul RORIC-LING <i>lunile 1-6</i>	323
II. GENERAREA SEMIAUTOMATĂ A SYNSET-URILOR ȘI CLUSTER-ELOR ROMÂNEȘTI DE TIP WORDNET	
Florentina Hristea <i>Asupra generării semiautomate a synset-urilor și cluster-elor de tip WordNet cu specială referire la limba română</i>	369
Theodor Hristea <i>Unele comentarii lingvistice privind rezultatele obținute</i>	409
Buletinul RORIC-LING <i>lunile 7-12</i>	415

III. O SPECIFICAȚIE TEORETICĂ PENTRU UN MODEL MORFOLOGIC AL LIMBII ROMÂNE

Theodor Hristea și Cristian Moroianu

Generarea formelor flexionare substantive și adjectivale

în limba română 443

Emil Ionescu

Premise ale unui dicționar morfologic electronic

al limbii române 461

Buletinul RORIC-LING

lunile 13-18 469

Editors' Foreword

This book encloses all papers authored by the members of the Romanian Regional Information Centre for Human Language Technologies (RORIC-LING), together with data samples and bulletins corresponding to the three virtual seminars that have been held every six months (2001-2003). It represents an attempt to help raise the awareness concerning some of the most advanced Human Language Technologies (HLT), as well as the possible scientific and industrial applications of the corresponding linguistic resources, both in Romania and in the entire Balkan region.

RORIC-LING is part of the BALRIC-LING project, funded by the European Commission (IST-2000-26454). The goal of RORIC-LING is that of building awareness in HLT primarily in Romania, a country still lacking true Language Engineering applications and HLT markets. The RORIC-LING information desk was open both to specialists and to nonspecialists, addressing subscribers coming from academic and research units, from software companies, as well as from other fields of activity.

Since HLT is a very broad field, RORIC-LING addresses only the following three topics:

- grammatical formalisms and their usage in the case of the Romanian language; corresponding tools for corpora annotation;
- semiautomatic generation of WordNet type Romanian synsets and clusters;
- a theoretical specification concerning a morphological model for Romanian.

The present book reflects the web site¹ that has been created within the framework of this project in order to facilitate communication with potential clients as well as quick dissemination of the information concerning all RORIC-LING topics. This web site contains all papers authored by the RORIC-LING specialists,

¹ <http://phobos.cs.unibuc.ro/roric>

accompanied by corresponding data samples and on-line demos, together with bulletins reflecting the virtual seminars that took place in connection with each of the RORIC-LING topics.

The project site itself is much richer than the contents of the present book. It includes comprehensive overviews, authored by specialists from ILSP (Greece) and Sheffield University (UK), which refer to the main BALRIC-LING topics. The same site is very rich in on-line demos that should be used by all visitors trying to get acquainted with a relatively new field.

Within this book the RORIC-LING topics are addressed bilingually, the same as in the project web page. The first part of the book is in English, while the second part represents the corresponding Romanian translation. Each of the two identical parts is organized according to the RORIC-LING topics. All materials have been taken directly from the RORIC-LING web site. As a result of a minimal editing effort, due to the time limits imposed by the project duration, and in order for the book to reflect the project web page as closely as possible, its style is not uniform. We ask the reader to accept our apologies for this inconvenience.

We hope this book will be of interest to all those involved in the field of HLT, but also to those who are not yet familiar with the field. The primary goal of this book is for it to act as an open invitation for its readers to access the project web page that, we hope, will have much more to offer.

The RORIC-LING team is greatly indebted to the European Commission for having encouraged this awareness effort. We would like to thank the European Commission for the importance it has attached to the RORIC-LING topics, as well as for having offered its full support. Special thanks are owed to Dr. Galia Angelova of the Bulgarian Academy of Sciences, Linguistic Modelling Department, the BALRIC-LING coordinator, for the continual support of our team within the framework of this project, as well as over the last years.

February 2003

Florentina Hristea

Marius Popescu

Objectives of BALRIC-LING

Dr. Galia Angelova

BALRIC-LING coordinator

RORIC-LING is part of the broader BALRIC-LING project, funded by the European Commission (IST-2000-26454).

The first main BALRIC-LING objective is to raise the awareness in the newly associated Balkan countries Bulgaria and Romania concerning the potential of the most advanced Human Language Technologies (HLT) and the possible scientific and industrial applications of the corresponding linguistic resources. It is very important to create such awareness in these countries, because they are on their way to full EC integration. However, examples of successful HLT marketable applications do not exist at all for Bulgarian and Romanian. Having been isolated for many years from the broad West European scientific exchange of language engineering ideas and practice, and having to deal with structurally different languages, the very few advanced HLT-groups from both countries still cannot gather by themselves a critical mass of informed specialists who could raise the quality of Language Engineering (LE) applications for the newly emerging HLT markets in Bulgaria and Romania.

Since HLT is a rather broad field, BALRIC-LING focuses on four topics only:

- word-centered linguistic resources and annotations;
- corpora and tagging;
- relevant supporting tools;
- possible advanced HLT usages of the considered resources.

To meet the target of building awareness in Bulgaria and Romania, BALRIC-LING has aimed at the realization of some main initiatives, to which we shall refer in what follows.

One such initiative was the development of Regional Information Centers (RICs) in BULgaria and ROmania (BULRIC and RORIC respectively). Most generally, these information centers represent Web sites with HLT tools descriptions, data samples, linguistic resources and prototypes of related supporting tools. The sites support an information desk, where specialists from the consortia have prepared comprehensive overviews of the four BALRIC-LING topics, and interested organizations and individuals have posed their queries, relevant to the RICs thematics. Documents in English as well as in Bulgarian and Romanian correspondingly for each country, are available on the site. We consider this the most appropriate way for the materials to reach the public and interested companies and research groups in the respective Balkan states. In addition to this, the RICs sites contain information about pertinent conferences, workshops and summer schools organized in Europe and closely related to BALRIC-LING topics, since information about such events is extremely scarce in Bulgaria and Romania.

Virtual seminars, based on the queries received at the RIC centers support desks, have been held every six months. Subscribers to especially organized mailing lists with broad coverage have asked questions concerning all materials exposed at the corresponding RICs. Specialists from the consortia have prepared answers that were mailed to all seminar subscribers once every six months (i.e. 3 times during the network duration). These seminars have facilitated raising the awareness, as well as the distribution of expertise from the more informed academic units to interested industrial organizations in Bulgaria and Romania. Biannual virtual bulletins in Bulgarian and Romanian respectively have been directed to subscribers of virtual seminars and have allowed for broad dissemination of BALRIC-LING initiatives especially among linguistic units and among individuals in the countryside of Bulgaria and Romania.

The second main BALRIC-LING objective is to help Balkan research groups in becoming better prepared for scientific co-operation at European level. ILSP (Greece) and Sheffield University (UK) have shared their rich experience and

practice in conducting successful research at both European and national dimensions. The Regional Information Centers in Bulgaria and Romania have contributed to the dissemination of HLT ideas among software companies which are interested in further development of advanced HLT applications for Bulgarian and Romanian.

One of the ways to facilitate the formation of future project consortia is the exchange of information about existing formats and standardization of the internal representation of some available resources of all partners. Being prepared in unified formats, those resources can be smoothly integrated for simultaneous use in multilingual applications and further developed. All standardization requirements, set by BALRIC-LING, are publicly available, so Balkan research groups and interested software companies may refer to them as guideline.

BALRIC-LING aims at the standardization of two formats of internal representation:

- standardization of formats for encoding of monolingual and parallel corpora;
- standardization of formats for the internal representations of grammatical dictionaries in the three Balkan countries.

The compact BALRIC-LING configuration allows for in-depth acquaintance with all details exchanged via narrower communication among partners. BALRIC-LING participants from EU-member countries will prepare an overview with evaluation of the progress of the awareness and dissemination tasks in the newly associated Balkan countries Bulgaria and Romania.

I

GRAMMATICAL FORMALISMS AND THEIR USAGE IN THE CASE OF THE ROMANIAN LANGUAGE. CORRESPONDING TOOLS FOR CORPORA ANNOTATION.

A Dependency Grammar Approach to Syntactic Analysis With Special Reference to Romanian

Florentina Hristea and Marius Popescu

1 Introduction

At the heart of sentence structure are the relations among words, no matter if by these relations we mean the possible grammatical functions or the links which bind words into larger units (like phrases, for instance). The *dependency grammar* approach to syntactic analysis takes into consideration the latter, viewing each word as *depending* on another word which links it to the rest of the sentence. Unlike generative grammars therefore, *dependency grammars* (DG), are not based on the notion of constituent but on the direct relations existing among words.

The relation between the *dependent* word and the word on which it depends (the *head*) is at the basis of DG. The syntactical analysis of a sentence means, from the point of view of DG, the description of all dependency relations (between the head and the dependent) which occur among all words of the sentence.

One way of graphically representing dependency relations (and, therefore, the DG syntactic structure of a sentence) is that of annotating the sentence with arrows which point *from dependent to head*, each arrow having a tag which indicates the type of relation it represents. Examples of this type of graphic representation of a sentence's syntactic structure can be seen in §4.

2 Dependency relations

A variety of dependency relations may exist among the words of a sentence if no restrictions are specified. The role of dependency grammars is mainly that of specifying the restrictions which the dependency relations should meet so that the structure which they define is linguistically correct.

No matter what language it means to define, any dependency grammar should take into account a number of linguistically motivated general principles. These principles are:

- Any word should depend exactly on one other word (the head), with the exception of the main predicate in the sentence which depends on no other word.
- Several words may depend on the same head.
- If the relations between the dependent and the head are represented by an arch, from the head to the dependent, then these arches should not intersect, and the oriented graph which is thus formed should not contain cycles.

Besides these general principles a dependency grammar can also specify what relations are allowed among various words, according to the part of speech they represent or according to other criteria. For instance, a dependency grammar could specify that a verb may not depend on a noun, or which part of speech may an adjective depend on etc.

The dependency relations which define the structure of a sentence may be described by means of the *dependency structure* and of the *type* of the dependencies. The *dependency structure* will specify, in the case of each word, what other word it depends on. The *dependency type* will specify, in the case of each dependency, its type.

Formally, the dependency relations may be described as follows:

Let \mathcal{W} be a finite set of words (the vocabulary) upon which sentences can be formed. A sentence will be designated by the sequence of words

$$w_1, w_2, \dots, w_n$$

We shall denote by w_0 a special word called *BOS* which denotes the beginning of the sentence and by w_{n+1} the special word *EOS* which denotes the end of the sentence.

Let \mathcal{T} be a finite set of *tags* called parts of speech. They will represent the part of speech (noun, verb, adjective, etc.) to which the words of vocabulary \mathcal{W} can belong.

Let also \mathcal{D} be a finite set of *dependency types* (subject, attribute, determiner, etc.).

The syntactical structure of a sentence

$$w_1, w_2, \dots, w_n$$

will be a structure (S, D) , where S will be named the *dependency structure*, and D will be named the *dependencies type*.

The *dependency structure* S is, in turn, a structure (T, P) , where $T = t_1, t_2, \dots, t_n; t_i \in T \forall 1 \leq i \leq n$ is a sequence of tags t_i designating the part of speech to which the word w_i belongs, while $P = p_1, p_2, \dots, p_n; p_i \in \{1, 2, \dots, n, n+1\} \forall 1 \leq i \leq n$ is a sequence of numbers which specify for each word w_i its parent (the word on which it depends). The parent of the word w_i will be the word w_{p_i} . Since there exists a word w_h which depends on no one (the main predicate of the sentence), it will be considered that the parent of this word is $w_{n+1} = EOS$, therefore $p_h = n+1$.

The *dependencies type* D is a function $D : \{1, 2, \dots, n\} \rightarrow \mathcal{D}$ where $D(i) = d$ represents the type of dependency between the words w_i and w_{p_i} .

The general principles which the dependency relations should meet can be "translated", according to this formalism as follows:

For any sentence

$$w_1, w_2, \dots, w_n$$

- $\exists h, 1 \leq h \leq n$ so that $p_h = n+1$ and $\forall i, 1 \leq i \leq n, i \neq h \ p_i \in \{1, 2, \dots, n\}$ and $p_i \neq i$.
- $\forall 1 \leq i < j \leq n$
 - If $p_i < i$ then $p_j \leq p_i$ or $p_j \geq i$.
 - If $i < p_i \leq j$ then $p_j \leq i$ or $p_j \geq p_i$.
 - If $p_i > j$ then $i \leq p_j \leq p_i$.

(The above mentioned conditions are the formal expression of the fact that the arches defined by the dependency relations should not intersect).

- The dependency structure (T, P) defines a graph $G = (V, E)$, where $V = \{1, 2, \dots, n, n+1\}$, and $E = \{(i, p_i) | 1 \leq i \leq n\}$. This graph should have no cycles.

For example, when considering the set of tags $\mathcal{T} = \{NN, VBD, DT, IN\}$ and the set of dependencies type $\mathcal{D} = \{subj, attr, det, pred\}$, then the *dependency relations* for the sentence

i	1	2	3	4	5	6	7
w_i	the	price	of	the	stock	fell	EOS

are described by the structure (S, D) ; $S = (T, P)$ where

$T = DT, NN, IN, DT, NN, VBD$

$P = 2, 6, 2, 5, 3, 7$

$D(1) = det, D(2) = subj, D(3) = attr, D(4) = det,$

$D(5) = det, D(6) = pred$

Or, more compactly:

i	1	2	3	4	5	6	7
w_i	the	price	of	the	stock	fell	EOS
t_i	DT	NN	IN	DT	NN	VBD	EOS
p_i	2	6	2	5	3	7	0
d_i	det	subj	attr	det	det	pred	EOS

One should notice that the dependency relations thus defined are in accordance with the general principles.

Once the dependency relations are formally defined, a dependency grammar is a structure (R, C) , with R being a set of restrictions $R \subset T \times T \times \mathcal{D} \cup T \times W \times \mathcal{D} \cup W \times T \times \mathcal{D} \cup W \times W \times \mathcal{D}$ and C being a set of requirements, $C \subset T \times T \times \mathcal{D} \cup T \times W \times \mathcal{D} \cup W \times T \times \mathcal{D} \cup W \times W \times \mathcal{D}$.

A *syntactical structure* (S, D) ; $S = (T, P)$ relative to a sentence w_1, w_2, \dots, w_n is correct from the point of view of grammar (R, C) if

- $\forall 1 \leq i \leq n, (t_i, t_{p_i}, D(i)) \in R \vee (t_i, w_{p_i}, D(i)) \in R \vee (w_i, t_{p_i}, D(i)) \in R \vee (w_i, w_{p_i}, D(i)) \in R$
(the restrictions are met)
 - $\forall 1 \leq i \leq n$
 - if $\exists (t_i, t, d) \in C$ then $\exists 1 \leq j \leq n$ so that $p_j = i, t_j = t, D(j) = d$.
 - if $\exists (t_i, w, d) \in C$ then $\exists 1 \leq j \leq n$ so that $p_j = i, w_j = w, D(j) = d$.
 - if $\exists (w_i, t, d) \in C$ then $\exists 1 \leq j \leq n$ so that $p_j = i, t_j = t, D(j) = d$.
 - if $\exists (w_i, w, d) \in C$ then $\exists 1 \leq j \leq n$ so that $p_j = i, w_j = w, D(j) = d$.
- (the requirements are met)

The restrictions require that only certain relations among words should be considered valid, according to the words or their types, while requirements allow certain words or types of words to call for the existence within the sentence of other words or types of words which should satisfy certain relations.

It is useful that the process of syntactical analysis take place in two stages. During a first stage one should find the dependency structure (T, P) , while during a second stage the dependencies type D should be established.

After the dependency structure is known, in establishing, for each pair of words (w_i, w_{p_i}) , the type of the corresponding dependency relation, one must take into account, among other things, the grammatical function, as well as the fact that this grammatical function is always the function of the dependent in relation to the other word.

In establishing the most frequent types of dependencies for Romanian we have, in most cases, considered the syntactic function (as in classical syntactic analysis) of the *dependent*. In those cases where the dependent word is a preposition or a coordinating conjunction, no matter what head word is involved, the dependency type was established according to the (classical) syntactic function of the element (word) introduced by that preposition or conjunction. In other, fewer cases, the morphological characteristics of the dependent were taken into account, as can be seen in Table 1. (*Example*: the dependent word is an indefinite article and the dependency relation is called "indefiniteness relation"). The dependency type was given by the head word only in those cases where the head is represented by a preposition or a coordinating conjunction (see Table 1). These dependencies could be further refined according to the grammatical category of the dependent word (noun, pronoun, verb, adjective etc.), giving birth to relations such as "adjectival preposition government", "nominal preposition government" etc. However, such a refinement was not of interest to us because large corpora was not available, and our approach remained that presented in Table 1.¹

3 Dependency relations in Romanian

Table 1 lists some of the most frequent types of dependency relations which were found as occurring in the Romanian language. We must specify that the framework corresponding to this table² includes only scientific texts in the restricted field of chemical technology. The dependency types were established as mentioned in §2., while the table classifies them according to the head word.

TABLE 1

HEAD WORD	DEPENDENT WORD	DEPENDENCY RELATION TYPE	ABBREVIATION
verb	preposition	adverbial	ADV.
verb	preposition	prepositional object	P.O.
verb	preposition	agent phrase	A.Ph.
verb	preposition	direct object	D.O.
verb	participle	direct object	D.O.
verb (participle)	auxiliary	auxiliary relation	AUX.
verb (infinitive)	"a"	infinitival relation	INF.
verb	coordinating conjunction	direct object	D.O.

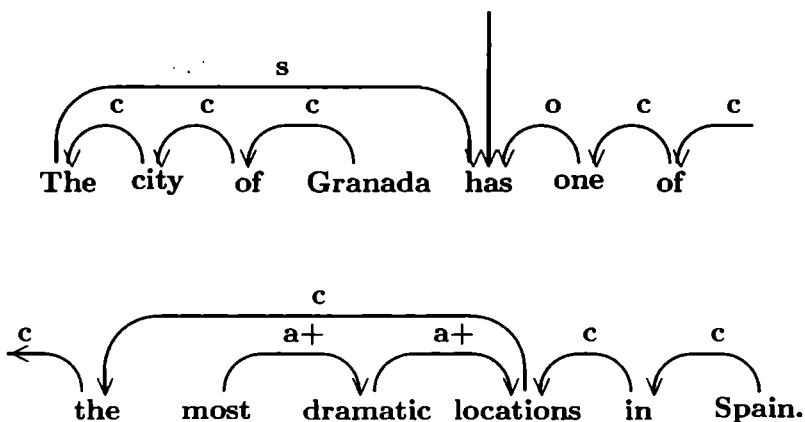
¹In Table 1, whenever referring to a head word of type verb, unless otherwise specified, the verb can be either a finite form or a non-finite form.

²Table 1 will be enlarged and soon published on the web according to newspaper articles being currently annotated by Romanian linguists.

HEAD WORD	DEPENDENT WORD	DEPENDENCY RELATION TYPE	ABBREVIATION
verb	adverb	adverbial	ADV.
verb	reflexive pronoun	reflexive relation	REF.
verb	noun	subject	S.
verb	pronoun	subject	S.
verb	numeral	subject	S.
verb	non-finite form	subject	S.
verb	noun	direct object	D.O.
verb	pronoun	direct object	D.O.
verb	numeral	direct object	D.O.
verb	non-finite form	direct object	D.O.
verb	noun	indirect object	I.O.
verb	pronoun	indirect object	I.O.
verb	noun	predicative	PRED.
verb	adjective	predicative	PRED.
verb	adverb	predicative	PRED.
verb	pronoun	predicative	PRED.
verb	numeral	predicative	PRED.
verb	non-finite form	predicative	PRED.
preposition	noun	preposition government	PREP.
preposition	coordinating conjunction	preposition government	PREP.
preposition	demonstrative pronoun	preposition government	PREP.
preposition	infinitive form	preposition government	PREP.
coordinating conjunction	noun	conjunctive relation	CONJ.
coordinating conjunction	verb	conjunctive relation	CONJ.
coordinating conjunction	verbal adjective	conjunctive relation	CONJ.
verbal adjective	preposition	agent phrase	A.Ph.
verbal adjective	noun	indirect object	I.O.
verbal adjective	preposition	adverbial	ADV.
verbal adjective	coordinating conjunction	indirect object	I.O.
verbal adjective	adverb	adverbial	ADV.
adjective	preposition	prepositional object	P.O.
adjective	adverb	relation of comparison	COMP.
adjective	demonstrative article	relation of comparison	COMP.
noun	indefinite article	indefiniteness relation	IND.
noun	preposition	nominal modifier (or attribute)	N.M.
noun	coordinating conjunction	nominal modifier	N.M.
noun	noun	appositive modifier	A.M.
noun	noun	nominal modifier	N.M.
noun	adjective	adjectival attribute	A.A.
noun	pronoun	pronominal modifier	P.M.
adverb	adverb	relation of comparison	COMP.
adverb	demonstrative article	relation of comparison	COMP.

4 Examples and comments

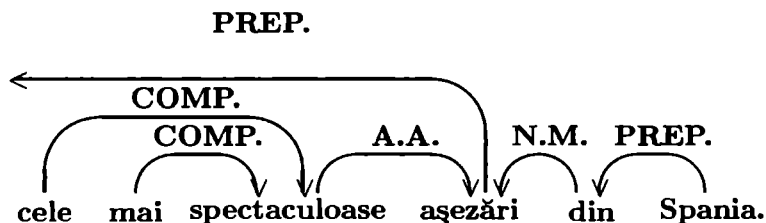
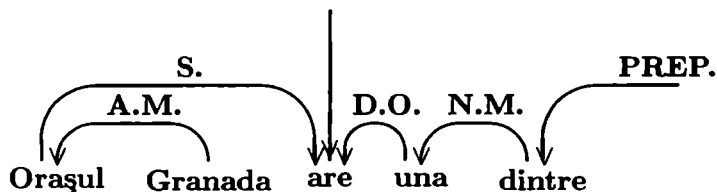
To better notice the difference between our approach to DG syntactic analysis (with special reference to Romanian) and the classical DG theory existing mainly for English and known mainly as word grammar (WG) theory (see [1], [2], [3]), let us consider the DG syntactic structure of the same sentence written in both languages. R. Hudson performs the syntactic analysis of the following English sentence as shown below:



Let us notice that, besides the classical dependencies of type "subject" (s), "object" (o) and "complement" (c), which we have refined, for the Romanian language, according mainly to syntactic function as in Table 1, a dependency relation of type "pre-adjunct" (a+) also occurs. (In WG theory for English a dependency is labelled "a" for "adjunct" to suggest that it is not "expected" by the head. This type of labelling is probably suggested by the definition of "adjunct", as in classical English grammar "a word or word group that qualifies, amplifies, or completes the meaning of another word or other words and is not itself one of the principal structural elements in its sentence".³ As in other cases, our adapted theory for Romanian either includes this type of dependency among those established according to syntactic function, or provides new types of dependency relations, such as the so-called "relation of comparison", which can be found in Table 1 and which also occurs in the translation of the previous sentence.)

The corresponding Romanian translation is a sentence having the following syntactic structure (according to Table 1):

³For this definition see "Webster's Third New International Dictionary", Springfield, Massachusetts, U.S.A., 1976



Acknowledgements

The authors would like to express their deepest gratitude to Prof. Dr. Theodor Hristea and to Assistant Cristian Moroianu of the University of Bucharest, Department of Romanian Language for providing the necessary linguistic guidance.

References

- [1] **R. Hudson** 1984. *Word Grammar*.
Oxford: Blackwell
- [2] **R. Hudson** 1990. *English Word Grammar*.
Oxford: Blackwell
- [3] **R. Hudson** 1998. *English Grammar*.
London: Rontledge
- [4] **I. A. Mel'čuk** 1987. *Dependency Syntax: Theory and Practice*.
Buffalo: SUNY Press
- [5] **L. Tesniere** 1959. *Elements de syntaxe structurale*.
Paris: Klincksieck

Dependency Grammar Annotator

Marius Popescu

Introduction

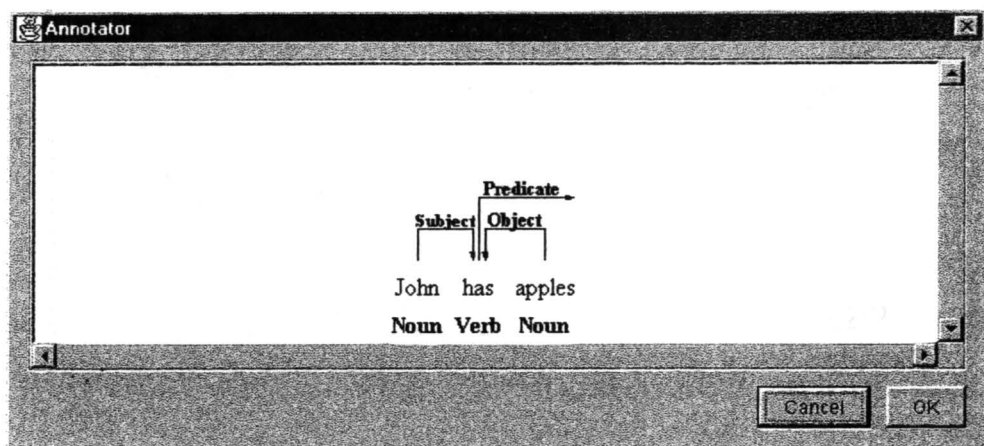
What is DGA

Dependency Grammar Annotator (**DGA**) is a tool conceived in order to facilitate the syntactic annotation of texts (of a corpus) within the formal framework of Dependency Grammars.

According to EAGLES¹: "syntactic annotation is the practice of adding syntactic information to a corpus by incorporating into the text indicators of the syntactic structure: eg. labeled bracketing, or symbols indicating dependency relations between words". Although very useful in practice (testing various grammatical theories, automatic acquisition of grammars etc.), such corpora are very costly since the operation of syntactic annotation is a great consumer of time and effort on the part of those performing it. **DGA** has been designed in order to minimize the human effort necessary during the process of corpus creation.

DGA is an easy to use graphical interface which allows the efficient creation and manipulation of syntactic structures. Since the used formalism is that of Dependency Grammars, the syntactic structures in this case consist of the dependency relations formed within the words of a sentence, labeled with the corresponding parts of speech, and of the grammatical relations existing among these words. Traditionally, the dependency relations are indicated by means of arches which link the dependent word to that word which it determines, these arches being labeled with the name of the relation existing between the linked words. Such a graphical representation (being in conformity with the EAGLES recommendations) is used by **DGA** as the base of the annotation operation.

¹ Expert Advisory Group on Language Engineering Standards
(<http://www.ilc.pi.cnr.it/EAGLES/home.html>)



During the entire annotation process the user acts directly upon this graphical representation. As a consequence, besides the advantage of convenient usage, the accuracy of annotation increases, since the user receives an immediate graphical feedback regarding any changes performed in the syntactic structure. Operating upon the syntactic structure is extremely easy and intuitive: in order to create a dependency relation only two mouse clicks are necessary (on the two words involved in the relation which is being created). For labeling a word with the corresponding part of speech or for establishing the type of a dependency relation only one mouse click, followed by the selection of the label from an appropriate list, is needed. **DGA** therefore allows a quick text annotation.

In our view **DGA** responds to the requirements which Marcus et al. have identified as being significant regarding the annotation process:

- **Accuracy** - since one operates directly with the graphical representation and because of the immediate graphical feedback which **DGA** offers to the user.
- **Speed** - creation and handling of the dependency relations are very quick operations (by means of the mouse).
- **Consistency** - the sets of parts of speech and dependency relations are established by the user after which, in order for them to be used, it is only necessary to select them from various lists.

Features

- **Usage easiness**: the fact that the user operates directly with the graphical representation induces great easiness in using **DGA** and highly increases speed (for more details see What is DGA).
- **Portability**: **DGA** has been written in Java 2. Being a pure Java application, **DGA** can run practically on any platform / under any

operating system for which a Java 2 runtime environment (JRE) exists. (It has been tested for Windows 95/98/NT and Linux systems). Since it uses the *pluggable look and feel* technology, from the point of view of the interface, **DGA** will behave as a native application relatively to the platform on which it runs, the user already being familiar with the basic items of the interface, such as menus, buttons, standard dialog boxes etc.

- **Conformity with up-to-date standards:** **DGA** is designed according to the EAGLES recommendations concerning syntactic annotation. The annotated texts are saved in XML format, as representing the standard in data description adopted by the linguistic community as the standard way of representing corpora. Although a standard set of XML tags for syntactic annotation does not exist yet, as is the case for morpho-syntactic annotation (XCES²), **DGA** uses a minimal set of tags inspired by XCES. Thus, the XML files produced by **DGA** can be easily transformed, by means of XSLT, into XML files which are based on a different vocabulary (tag set) meeting the requirements of the user or being in conformity with a future standard. For more technical details see The XML format used by **DGA**.
- **Flexibility:** besides the fact that syntactic analysis must have the form of dependency relations, **DGA** does not impose any other restrictions upon the user. The latter may easily define and at any time modify his own parts of speech and dependency relations sets, which will be used in annotation.

User Guide

Installation

Requirements:

1. In order to run, **DGA** requires the Java 2. Therefore the operating system of the computer on which **DGA** is to be installed must be one for which an implementation of Java 2 exists (Windows 95/98/ME/NT/000, most Unix versions, MacOS X).
2. The system on which **DGA** is to be installed must be sufficiently powerful (processor speed, memory). In the case of a PC, a minimum of 133 MHz and 32M RAM are necessary.

² XML Corpus Encoding Standard (<http://www.cs.vassar.edu/XCES>)

Installation:

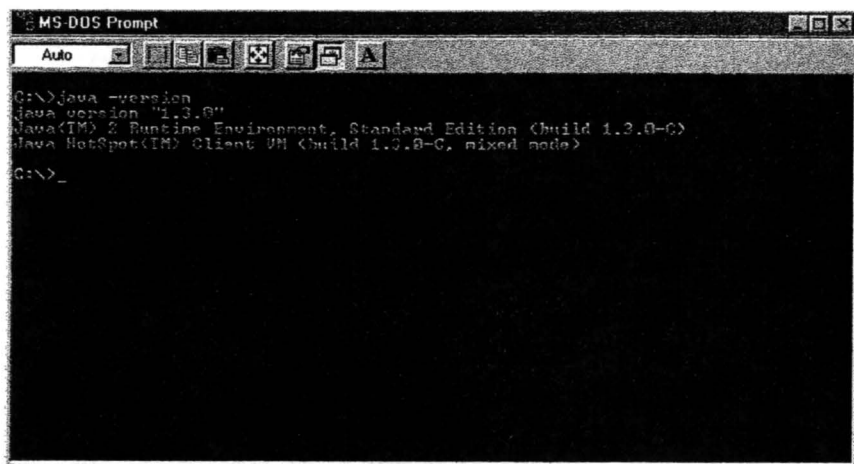
1. Install Java 2 on your system. If Java 2 is already installed, skip this step. Corresponding to Windows, Linux, Solaris platforms, the necessary installation kits can be found at java.sun.com. The entire JDK development kit or only the JRE runtime environment can be installed. Using version 1.3 or a more recent version is recommended.
2. Make sure that the PATH variable contains the path to the `java` executable. Corresponding to Windows 95/98/NT this can be achieved by including in the `autoexec.bat` file a line of the following form:

```
SET PATH=c:\path; %PATH%
```

where `c:\path` is to be replaced with the actual path leading to the directory where the `java` executable is placed. Following this operation (and restarting your computer), as a result of the command:

```
java -version
```

the answer of your system should look like:



```
MS-DOS Prompt
Auto
C:\>java -version
java version "1.3.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.0-G)
Java HotSpot(TM) Client VM (build 1.3.0-G, mixed mode)
C:\>_
```

no matter which directory the command was issued from.

3. Unzip the archive `dga.zip` and place its content wherever you wish within the existing directory structure.
4. In order to run `DGA`, from command prompt (MS-DOS prompt), from directory `DGAnnotator` (in order to make `DGAnnotator` your working directory type `cd path\to\DGAnnotator`), issue the command:


```
java -classpath .\dga.jar;crimson.jar;xalan.jar;jaxp.jar DGAAnnotator
```

Alternatively you can run the file **dga.bat** from the same directory. In order to facilitate future work sessions you can create a shortcut to the file **dga.bat** (which is to be placed on the desktop). Corresponding to the shortcut properties set the working directory to directory **DGAAnnotator**, and as icon choose the file **dga.ico** from this directory. At this point the DGA application



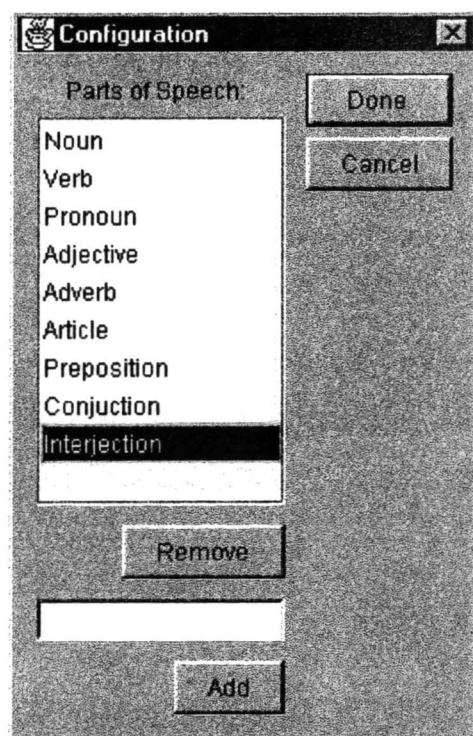
can run as a result of a double click on the icon

Remark: The above directions have concentrated especially on installation under the Windows operating system. In order to install and to run **DGA** on any system, in essence, you must be able to install the Java 2 environment on that system, to unzip the archive **dga.zip**, and then to run the (Java) class **DGAAnnotator**.

Configuration

After installation, **DGA** provides the user with an initial set of parts of speech and of dependency relations. This first set is just for the sake of illustrating how **DGA** works and has no linguistic relevance. The user must define his own sets of parts of speech and of dependency relations respectively, according to the specific natural language involved. This configuration operation (deletion or addition of parts of speech and/or of dependency relations) can take place at any moment, not just at the beginning of the annotation process. Obviously, one expects to start annotation using a first set of parts of speech and of dependency relations, while, on the way, parts of speech and dependency relations can be added or deleted according to specific necessities.


In order to configure (delete/add) the parts of speech set, select the **Parts of Speech** command from **Configuration** menu. As a result, a dialog box like the following one will be activated:



In order to delete a specific part of speech, select it from the list (with one mouse click on it), and click on the **Remove** button. As a result, the chosen part of speech will immediately disappear from the list. In order to add a specific part of speech, enter its name in the field on top of button **Add** and click on this button. Instantly, this new part of speech will occur at the end of the list. These two actions can be performed any number of times and in whatever order is desired. When the list becomes the required one, click on button **Done**. The dialog box will disappear, while the new part of speech list becomes immediately available for annotation. Obviously, one can cancel the configuration operation at any moment, by clicking on button **Cancel**.

In order to configure (delete/add) the dependency relation set, select command **Dependency Relations** from **Configuration** menu. A dialog box identical with the above one will be activated, with the unique difference that the parts of speech list will be replaced by the dependency relations list. From then on, the operating mode is identical with the one used when adding and deleting parts of speech.


Opening a document for annotation

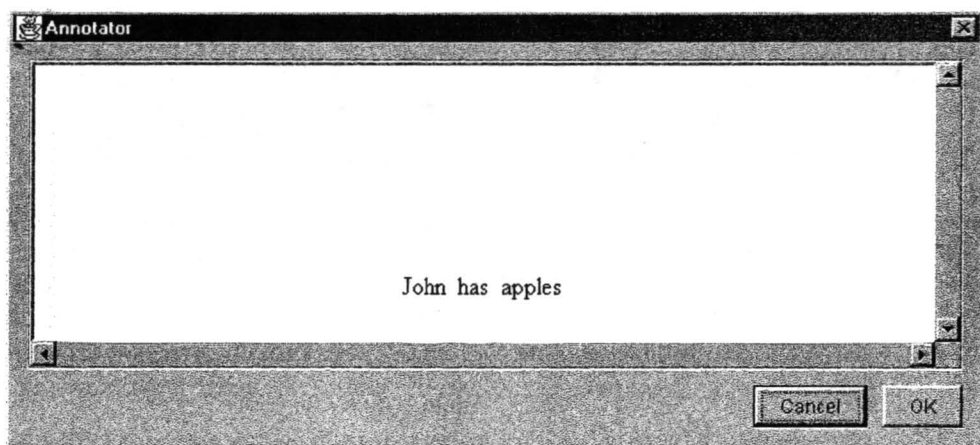
In order to annotate a text one should first open the file containing it. This file must be of type "text". The operation is performed by choosing the command **Open Text** from **File** menu or by clicking on button  from the tool bar. As a result, the

standard open dialog box (corresponding to the specific platform) will be activated. This dialog box will allow selection of the file to be opened. The content of the chosen file will be displayed within a window where the user may select sentences for annotation.

Remark: The user can not modify the contents of the displayed text file, the only action which is allowed being that of selecting parts of text (selecting a sentence for annotation).

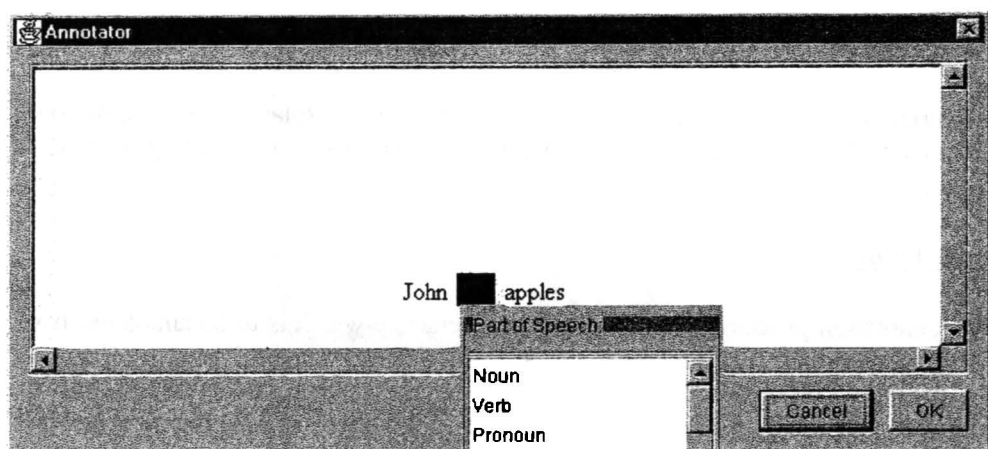
Annotation

The annotation process starts by selecting a sentence which is to be annotated from the displayed text. This operation is performed in the standard way, by dragging the mouse while pressing the left button, over the sentence (text) which must be selected. Once a fragment of text (sentence) is selected, the command **Annotate** from menu **Annotate** and the button  from the tool bar are enabled. When pressing this button or when choosing command **Annotate** from menu **Annotate** a dialog box will be activated. In this dialog box, within an editing field, is contained the text which has been selected. The user may edit the text here. Although the user can completely delete the text occurring in the editing field and can punch in a new one, this is not the reason for which this dialog box has been created. The reason for providing it is to allow the user to make non-significant changes within the sentence to be annotated. For instance, punctuation marks can be eliminated or separated from words if their annotation is also required, etc. Once the desired modifications are all performed, the annotation process itself can start when clicking on button **Annotate**. The selected (and maybe modified text) will be displayed within a new window where all annotation operations will be performed.

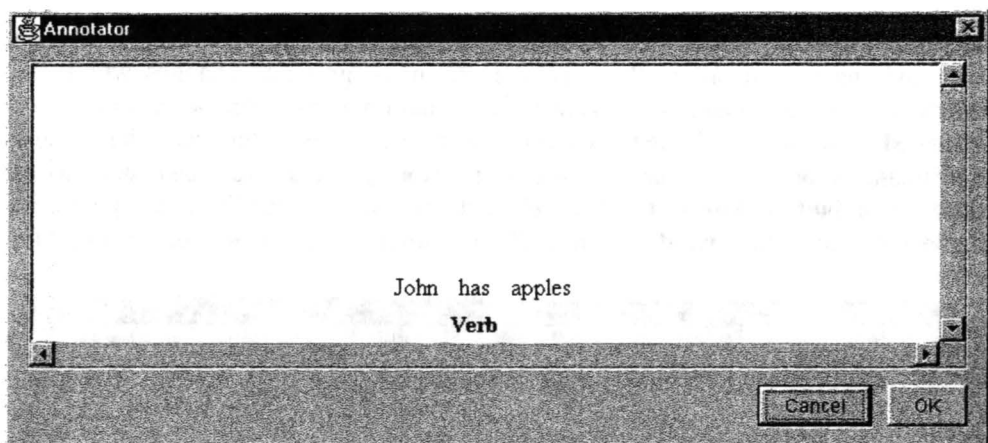


In order to establish the part of speech of a word, perform a right click on that specific word. The word will be marked by red coloring and underneath it a

contextual menu will be activated. This contextual menu contains the parts of speech list, (that established by the user see Configuration).

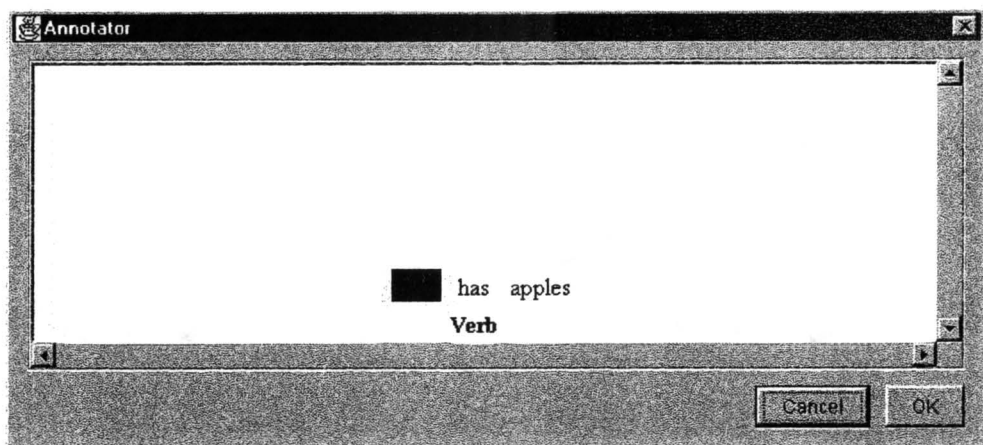


Select (click on) the desired part of speech and it will immediately occur underneath the word corresponding to which this action has been performed.

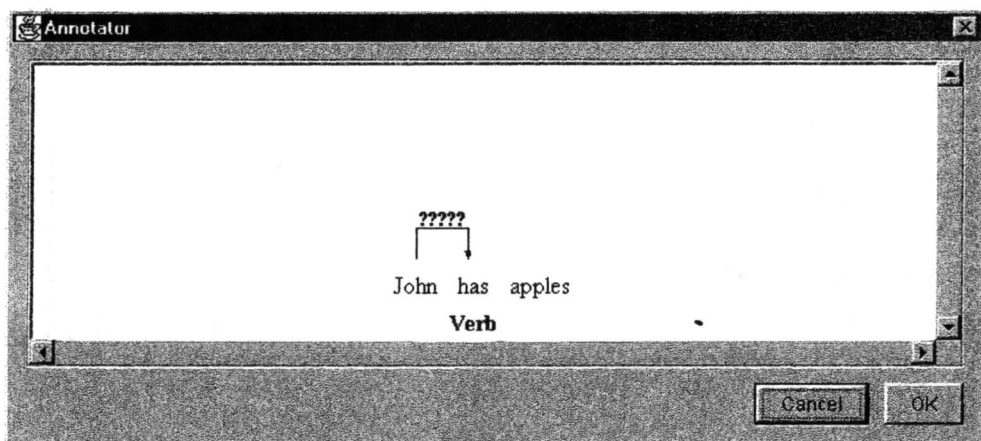


If the modification or deletion of a part of speech already associated to a word is required, performing a right click on that word or on its associated part of speech will again open the contextual menu containing the parts of speech list. When selecting a new part of speech it will immediately occur by replacing the previous one. If, within the contextual menu, click on the **Delete Part of Speech** button is performed, the part of speech corresponding to the word will be deleted and no part of speech will be associated to this word any longer.

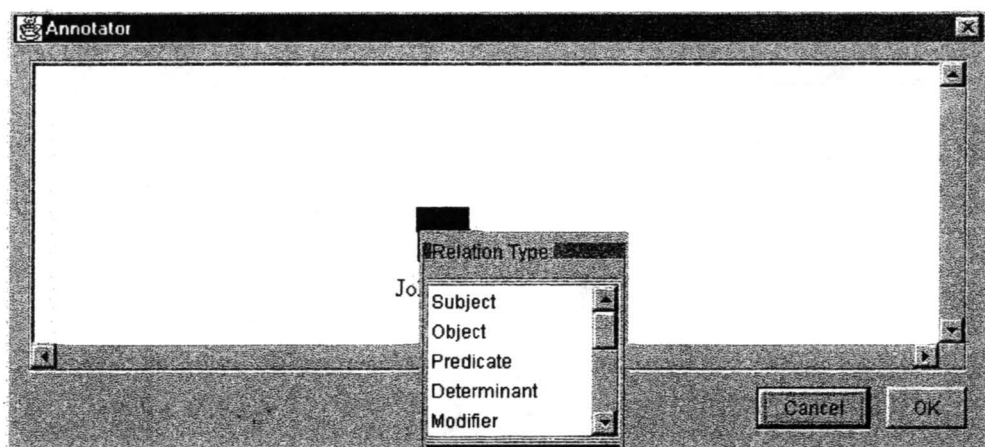
In order to create a dependency relation between two words, one first performs click (left button) on the *dependent* word (the one which determines and from where the arch starts). This word will be marked by blue coloring.



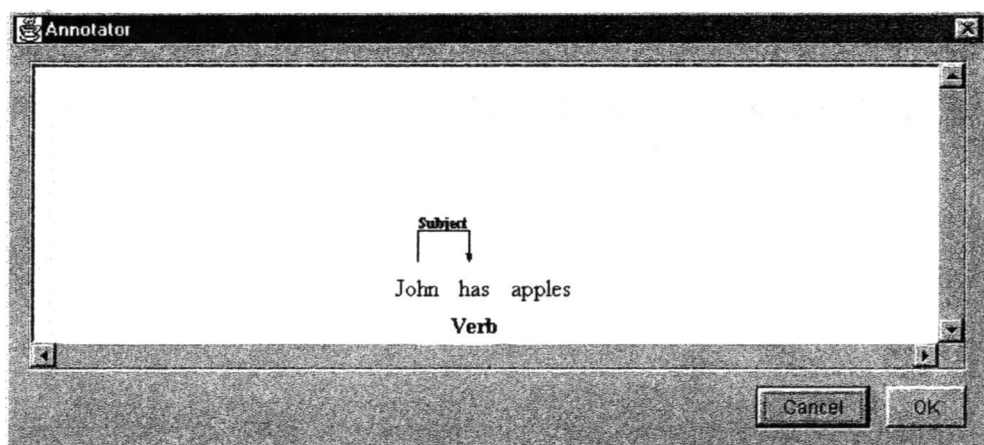
Next, click on the *head* word (the determined one, the one where the arch arrives) will be performed. An arch from the dependent word to the head word will instantly be created. This arch will be labeled with the sequence "?????", which indicates the fact that the type of the newly created dependency relation has not yet been established.



In order to establish the type of a dependency relation one must perform a right click on the sequence "?????" which occurs on top of the arch representing the relation. A contextual menu from where the type (established by the user, see Configuration) of the dependency relation can be chosen will open.

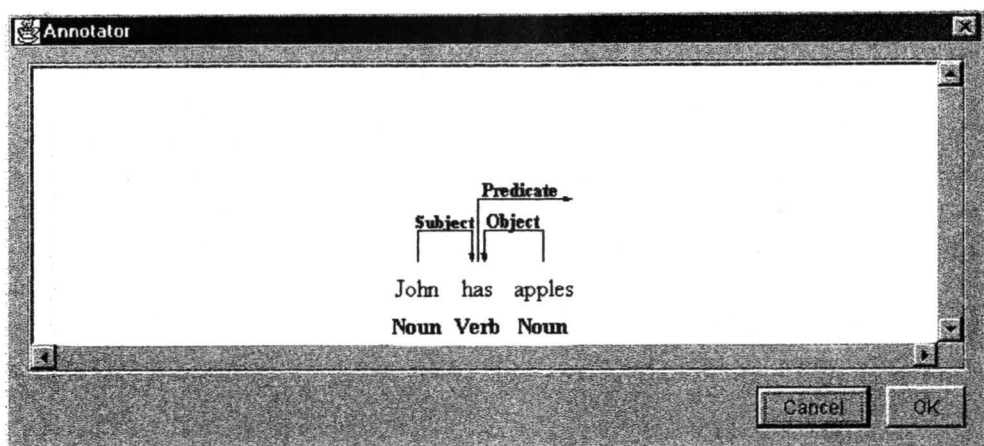


When selecting the relation type from the list (just like when establishing the part of speech), this type will immediately occur on top of the arch representing that relation.



Again just like in the case of the parts of speech, when the modification of the type of a dependency relation is required, one must perform a right click on that relation type and the contextual menu containing the types of relations list will be activated. By selecting a different type of relation, the new type will immediately replace the former one. When clicking on button **Delete Relation** of this contextual menu, the entire dependency relation (namely the corresponding arch) will be deleted.

According to the dependency grammar formalism, each word of a sentence must depend on (should determine) another. The unique exception is represented by just one word, considered a main word, the head of the entire sentence (usually the main verb), which depends on no other word. This fact is graphically indicated by means of an arch like the one attached to verb **has** in the following figure.



Specifying the fact that a word is the head of a sentence is achieved as follows: click on that word is performed and the word is marked by blue coloring; a second click on the word will have as consequence indicating it as the head of the sentence, by means of an arch, as in the above figure. Establishing the type corresponding to this new arch is performed like in the case of all other relation types. **Beware!** When marking a word as representing the head of a sentence do not perform a double click on it. Perform a single click, wait for it to be marked by blue coloring and only then perform a second click. Since a sentence has a unique head word, **DGA** allows defining only one such word. Once a head word corresponding to a certain sentence has been established, the sequence of actions by means of which a head word is defined will have no effect whatsoever when performed within that same sentence.

The actions concerning establishing the part of speech of a word, creating a dependency relation and establishing the type of such a relation can be repeated in any order chosen by the user, and until the user considers the sentence to be properly and completely annotated. At that moment, by clicking on the **OK** button, the process of annotating a specific sentence ends. Obviously, one can cancel the annotation process at any moment by clicking on **Cancel**. If one clicks on **OK** and annotation of the sentence is not complete (in the sense that words to which parts of speech have not been assigned still exist or that each word does not depend on another one and not all dependency relation types are known), then the user is notified but still has the opportunity to save the incompletely annotated sentence.

Remark: Each time when a right click was mentioned, one referred to the operation of obtaining a contextual menu. This is the case for Windows and Unix (X Windows), but can differ in the case of other platforms (MacOS, for instance).

Saving data and exiting

The annotated sentences are stored in memory. They are saved in a file only when the user explicitly chooses the **Save annotation** command from the **File** menu. The effect of this action is the activation of a standard (relatively to the specific platform) dialog box where the user, after having selected the directory where the file containing the annotated sentences should be placed, enters the name corresponding to this file. In order to save the file one must click on button **Save** of the dialog box. The file name must contain the extension **.xml** (corresponding to the format in which data are saved). Within the same dialog box the user may select an already existing file, in which case data will be appended at the end of that file (appended to the already existing data). It is up to the user to make sure that the selected file contains annotated sentences as well, and not data of a different nature.

The xml files in which data are stored can be placed at any location within the existing directory structure but, in order to be useful, they require the **dga.dtd** file which is placed in the directory of the **DGAnnotator** application. It is therefore recommended that file **dga.dtd** should be copied in those directories where the xml data files are located.

Ending a working session takes place by selecting the **Exit** command from **File** menu, or by closing the main window of the application. If, when exiting, data (annotated sentences) which have not yet been saved still exist, the user is notified and has the opportunity to save them.

The bookmark system

In order to obtain a corpus one usually annotates a great number of texts. Since it is very possible that annotation of an entire text can not be completed during a single working session, **DGA** is equipped with a bookmark system, which allows the user to get back to a specific position in a specific text whenever necessary. Annotation can continue from that location on.

In order to mark a specific location within a text (whenever the corresponding text file is open for annotation), a portion of the text (representing the returning point) is selected and the command **Add Bookmark** of menu **Bookmarks** is chosen. Within the list of bookmarks contained in the **Bookmarks** menu a new bookmark will occur. This new bookmark will be labeled with the date and time of its creation.

In order for the user to access the exact location (text and position within text) indicated by a specific bookmark, he must select that bookmark within the

Bookmarks menu. If the selected bookmark indicates a position in the current file (the one currently open for annotation), then the text will be displayed (by scrolling), within the window, in such a way that the marked portion (towards which the bookmark points) is visible. If the chosen bookmark indicates a location within a file other than the current one, then the current file (text) will be closed and the file (text) towards which the bookmark points will be opened. The corresponding text will again be displayed within the window such that the marked portion (towards which the bookmark points) is visible. A bookmark can be used even if no file (text) is currently open, in which case the file towards which the bookmark points is automatically opened having its contents displayed.

In order not to overload the **Bookmarks** menu (and because a great number of bookmarks is not usually required), **DGA** limits the maximum number of bookmarks to 10. The user has the opportunity to delete those bookmarks which will no longer be used by selecting the **Remove Bookmark** command from menu **Bookmarks**. When doing this a dialog box containing the list of existing bookmarks will be activated. As a consequence of selecting a bookmark from this list and of clicking on the **Remove** button, the selected bookmark will be deleted.

Viewing and modifying an annotated text

DGA allows viewing and possibly modifying the annotations of previously annotated texts. In order to do this, one must choose the **Open corpus** command from menu **File**. As a result, a standard dialog box which enables selection of the file containing the annotated text (corpus) will be opened. The selected file must be a xml file, having a structure in accordance with the **DGA** format (see The XML format used by **DGA**).

The contents of the chosen file will be displayed within a window as follows: the text will be displayed as a set of sentences, each sentence occurring as a hyperlink (in a web browser). When clicking on a sentence, its annotation will be displayed, in the usual graphical form, within a dialog box. Annotation modifications can be performed here, all annotation operations being allowed (see the section referring to annotation).

Saving modifications is performed by selecting one of the commands **Save corpus** or **Save corpus as** from menu **File**. Closing the file is performed by means of the command **Close corpus** of the same menu.

The XML format used by DGA

The annotated texts are saved in XML format, as representing the standard in data description adopted by the linguistic community as the standard way of representing corpora. Although a standard set of XML tags for syntactic annotation does not exist yet, as is the case for morpho-syntactic annotation (**XCES**), **DGA**

uses a minimal set of tags inspired by XCES. Thus, the XML files produced by DGA can be easily transformed, by means of XSLT, into XML files which are based on a different vocabulary (tag set) meeting the requirements of the user or being in conformity with a future standard.

In order to illustrate the used set of tags, we present the following fragment of a xml file, representing the annotation of the sentence "John has apples" (see What is DGA).

```
<s>
  <tok>
    <orth>John</orth>
    <ordno>1</ordno>
    <ctag>Noun</ctag>
    <syn>
      <head>2</head>
      <reltype>Subject</reltype>
    </syn>
  </tok>
  <tok>
    <orth>has</orth>
    <ordno>2</ordno>
    <ctag>Verb</ctag>
    <syn>
      <head>4</head>
      <reltype>Predicate</reltype>
    </syn>
  </tok>
  <tok>
    <orth>apples</orth>
    <ordno>3</ordno>
    <ctag>Noun</ctag>
    <syn>
      <head>2</head>
      <reltype>Object</reltype>
    </syn>
  </tok>
</s>
```

Each sentence is marked by tag `<s> ... </s>`. Each word of the sentence, together with all information concerning its annotation, is marked by tag `<tok> ... </tok>`. Within this tag, the orthographic form, as it occurs in the annotated text, is marked by tag `<orth> ... </orth>`. Tag `<ordno> ... </ordno>` indicates the number of the word within the sentence (counting is performed starting from the beginning of the sentence). By means of tag `<ctag> ... </ctag>` the part of speech is specified, while tag `<syn> ... </syn>` marks the syntactic information. Within tag `<syn> ... </syn>` the head word is specified by means of its number within the sentence, this number

being marked by tag <head> ... </head>. The type of the dependency relation existing between the two words (the one to which the annotation belongs and the head word) is specified by means of tag <reltype> ... </reltype>.

Bibliography

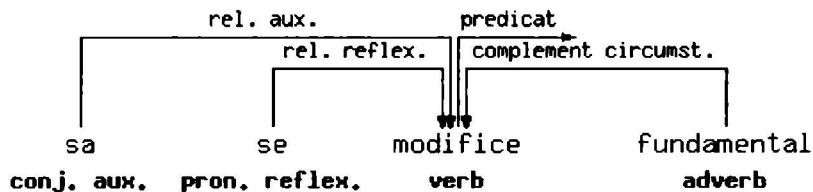
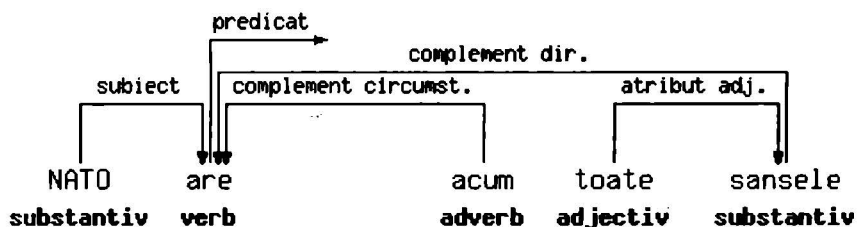
Hudson R. *English Word Grammar*. Oxford: Blackwell, 1990

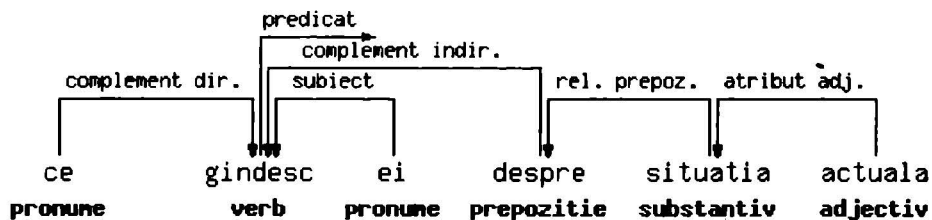
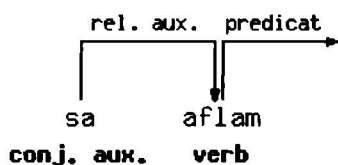
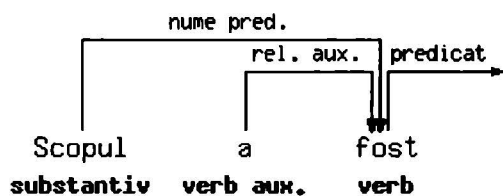
Marcus P.,Satornini B., Marcinkiewicz M. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313-330, 1993.

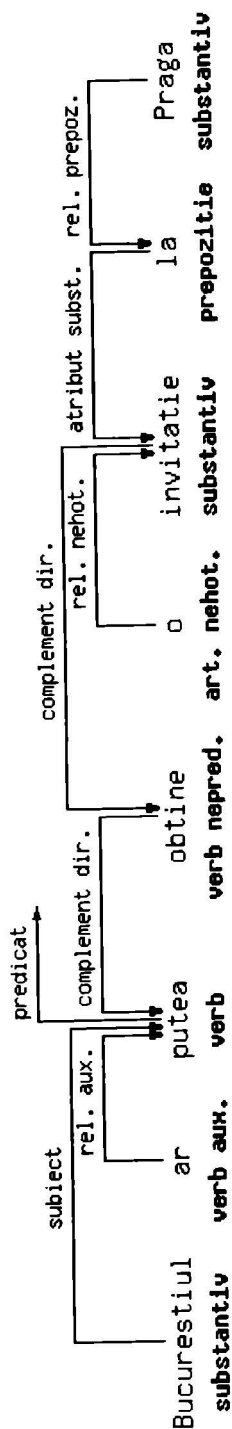
Tesniere L. *Elements de syntaxe structurale* Paris: Klincksieck, 1959

Appendix

Sample of Romanian sentences annotated with DGA







- A GENERAL PRESENTATION

Emil Ionescu

Abstract

This is a presentation of an influential framework in contemporary linguistics - Head-driven Phrase Structure Grammar (HPSG). The presentation is not exhaustive. Its aim is to point out to a reliable tool in explaining, analyzing and processing data of natural language.

The presentation has two main parts. In the previous one, we discuss aspects tight to the language in which the theory is expressed, as well as the main techniques and devices used in order to cope with phenomena of natural language. We thus present the attribute-value system of encoding linguistic information, the concept of unification, phenomena of coindexing (structure sharing), the notion of type and the type architecture. The previous section ends with a discussion about the notion of constraint.

The latter part concentrates on the main linguistic objects considered in HPSG: the word and the phrase. We review the main properties of both of them, that is, semantic, syntactic, pragmatic and morphological properties of words, and semantic, syntactic and pragmatic properties of phrases. The presentation ends with an inventory of the achievements of the theory.

Introduction

HPSG was formulated and proposed in two works of Carl Pollard and Ivan Sag (Pollard and Sag (1987) and Pollard and Sag (1994)), which became reference books in the field. The theory belongs to the family of unification grammars and it is based on constraints. Thus, HPSG resembles other formal grammars, such as Generalized Phrase Structure Grammar (GPSG), Lexical Functional Grammar (LFG), Tree Adjoining Grammar (TAG) or Categorical Grammar (CG).

1. Notation and Devices

1.1. Notation

In order to encode linguistic information, HPSG proposes a uniform notation, which is currently known as the system of the attribute-value matrices (AVMs). Prototypically,

an AVM contains an attribute (usually written in capitals) and a value (usually written in italics), as in AVM1, below:

AVM1

[*PHON* : <dog>]

AVM1 says that there is a linguistic object (a feature structure) characterized by the fact that it bears the phonological information indicated within the angular brackets¹. So, in AVM1, the attribute is PHON (from PHONOLOGY) and its value is the list of phonemes <dog>. The whole matrix observes the principle of the mathematical functions, because it is a law of correspondence between the attribute PHON and the sequence <dog>.

1.2. Structure Sharing

If two attributes have the same value, this may witness an essential identity of information (that is, an identity crucial for the well-formedness of a linguistic object), or on the contrary, a non-essential one (i.e. an identity which does not influence the well-formedness of the object). For instance, in the sentence:

(1) a. He loves himself/*herself/*ourselves.

the identity of gender, number and person between the personal pronoun **he** and the reflexive **himself** is compulsory for the grammaticality of (1a). However, the same identity is accidental in (1b), as far as the pronouns **he** and **him** are concerned:

(1) b. He loves him/her/us. ~

In HPSG, identities of type (1a.) are named ‘token identities’ (or ‘shared structures’) while those of kind (1b.) are ‘type identities’. The former (but not also the latter ones) are symbolized by coindexing:

AVM2

¹ Angular brackets in HPSG denote a list, that is, a set whose members are ordered.

$$\left[\begin{array}{l} \text{HEAD : verb} \\ \text{VAL : } \left[\begin{array}{l} \text{SUBJ : } \left[\begin{array}{l} \text{HEAD : noun} \\ \text{IND : } |1| \text{ref} \left[\begin{array}{l} \text{PERS : 3rd} \\ \text{NB : sin g} \\ \text{GEND : masc} \end{array} \right] \end{array} \right] \\ \text{COMPS : } \left[\begin{array}{l} \text{HEAD : noun} \\ \text{IND : } |1| \end{array} \right] \end{array} \right] \end{array} \right]$$

AVM2 gives the representation of the properties of the verb **loves** when its subject (SUBJ) is a personal pronoun (or a noun) and its complement (COMPS) is a reflexive. The feature HEAD indicates the part of speech. The feature VAL(ence) is a generalization over lexical and phrasal arguments of a certain kind, such as the subject, complements or specifiers. The feature IND(ex) is a semantic one and takes as value types of indexes². The indexes in AVM2 are of type *referential*. They must be identical, a fact which is noted through the coindexing of the values of IND for the subject **he** and the object **himself**, respectively.

If one gives the representation of the same lexical item for the environment (1b.), one may remark that coindexing between the referential index of the subject **he** and the one of the complement **him** is not compulsory any longer. This simply means that either one has to write again the information concerning the index, or that one has to leave it unspecified, like in AVM3:

AVM3

$$\left[\begin{array}{l} \text{HEAD : verb} \\ \text{VAL : } \left[\begin{array}{l} \text{SUBJ : } [\text{HEAD : noun}] \\ \text{COMPS : } [\text{HEAD : noun}] \end{array} \right] \end{array} \right]$$

It has to be added that in syntax, structure sharing is a valuable device in dealing with constructions which in the G&B Theory are treated by means of movement.

² In HPSG, the information of person, gender and number forms a single cluster called *index*. An index may be referential (like in (1a-b) or expletive like the index of *it* in "It is raining". This semantic information has to be carefully distinguished from the information carried by *affixes* of number, gender or person. This latter type of information is not semantic but – of course – morphological. Let us also notice that structure sharing between the index of **he** and the index of **himself** is noted by the tag *|1|*.

1.3. Types

Objects symbolized by italics are called **types**. Types may be atomic or non-atomic. An atomic type cannot be decomposed, because it bears no inner structure. A non-atomic type can. For instance, the object *acc(usative)*, which is one of the values of the attribute CASE, cannot be decomposed. It is therefore an atomic type. Boolean values + and – are also atomic. On the contrary, the type *noun* displays an inner structure. It is useful indeed to specify the case of a noun, and this specification shows that the noun may be 'analyzed'. Therefore, the type *noun* may label a whole matrix and it is a non-atomic type: *noun*[CASE:*acc*].

1.4. Type Hierarchies

Types are arranged in hierarchies, the principle of which is inheritance: a type *t_i* which has as supertype the type *t_j* inherits all the properties of *t_j*. Let us consider one of the most important and general types in HPSG: the object *sign*. What is of type *sign* may be either a word (*wd*) or a phrase (*ph*). One thus gets the following fragment of hierarchy:

H1: *sign*: *wd* or *ph*

The statement H1 expresses a disjunction: what is a sign is either a word or a phrase. Now, thanks to the inheritance principle, the properties of the type *sign* 'flow' to the subtypes *wd* and *ph*. Inheritance thus turns out to be a principle of economy, because due to it, there is no need to repeat the information that characterizes a supertype on its subtypes.

Hierarchies may also be multiple. Multiple hierarchies come from distinct criteria or classification (or dimensions) of the same types. Let us consider the situation of the type *ph*. A phrase may be regarded from at least two points of view: from the point of view of clausality and - borrowing a well-known term from L. Bloomfield - from the one of endocentricity³. According to the former perspective, a phrase may be a clause (*cl-ph*) or a non-clause (*n-cl-ph*). And according to endocentricity (headedness), a phrase may be headed (*hd-ph*) or non-headed (*n-hd-ph*). This further classification leads to the following statements:

H2: (a) *ph* (CLAUSALITY): *cl-ph*, or *n-cl-ph*
(b) *ph* (HEADEDNESS): *hd-ph* or *n-hd-ph*

³ The canonical term in HPSG is "headedness".

It may be now remarked that theoretically, a clausal phrase may be either headed or non-headed. The same holds for non-clausal phrases. One thus obtains maximal types which are in fact conjunctive types because they are obtained through conjunctions (usually noted \wedge): $cl-ph \wedge hd-ph$ or $cl-ph \wedge n-hd-ph$ or $n-cl-ph \wedge hd-ph$ or $n-cl-ph \wedge n-hd-ph$.

1.5. Unification

It is already apparent from the preceding sections that an AVM may contain several pairs attribute-value. In this case, the matrix displays a **conjunction** of pieces of linguistic information, or, in other words, unified linguistic information. This explains why HPSG is considered to belong to the family of the so-called **unification grammars** (Shieber (1986)), that is, grammars where linguistic correctness is accounted for by means of conjunctions of appropriate clusters of linguistic information. Such a conjunction is represented in AVM4, where the additional information is that the object described in AVM1 bears non-empty syntactic and semantic information ([SYNSEM: *nev*]):

AVM4

$$\left[\begin{array}{l} PHON : \langle dog \rangle \\ SYNSEM : nev \end{array} \right]$$

In AVM4, the type *nev* functions like a variable in a system of logic: it shows that the semantic and syntactic information of the described object may be left unspecified, in spite of the fact that this information is not empty.

1.6. Constraints

Not every conjunction of pieces of information displayed by an AVM represents a proper linguistic object. Let us consider AVM5 which 'stands for' the class of clausal phrases that all have in common the lack of agreement between the subject and the verb⁴:

AVM5

⁴ In AVM5, DTRS means 'daughters'. Daughters in AVM5 are a head-daughter (HD-DTR) and a subject-daughter (SUBJ-DTR)

$$\left[\begin{array}{l} \text{HEAD} : |1| \text{verb} \left[\text{IND} : \text{ref} \left[\begin{array}{l} \text{PERS} : 3\text{rd} \\ \text{NB} : \text{sin } g \end{array} \right] \right] \\ \\ \text{DTRS} : \left[\begin{array}{l} \text{HD} - \text{DTR} : \text{sign} [\text{HEAD} : |1|] \\ \text{SUBJ} - \text{DTR} : \text{sign} \left[\text{HEAD} : \text{noun} \left[\text{IND} : \text{ref} \left[\begin{array}{l} \text{PERS} : 3\text{rd} \\ \text{NB} : \text{plural} \end{array} \right] \right] \right] \end{array} \right] \end{array} \right]$$

Obviously, such representations are not welcome and they have to be ruled out. It is exactly this requirement that justifies the resort to constraints. Constraints are ‘privileged’ representations that capture relevant conditions of well-formedness on relevant objects. Typically, a constraint is an implicational statement in which the antecedent is a type and the consequent is an AVM. The statement has to be read: “if an object is of type t_i then it has to satisfy the conditions represented in the associated AVM”. Here is, for instance, the constraint on the type ‘phrase’ (ph):

$C(ph)$

$ph \Rightarrow$

$[DTRS : nev]$

What $C(ph)$ says is simply that if something is a phrase, then it has to have daughters ($[DTRS : nev]$). This is indeed the conventional wisdom about phrases: phrases are **signs** with phrasal inner structure (see above, H1).

But let us now consider the situation of direct object N(oun)P(hrase)s in Romanian, from the point of view of constraints. A peculiarity of direct object NPs is the fact that under given conditions they participate in the construction of phrases which are marked by the so-called preposition **pe**, (in fact, a marker of the direct object NPs in Romanian). In this case they form, along with the marker, a phrase which is defined as a special kind of head-marker phrase, symbolized here *hd-OBLmark-ph*. For instance, unlike English, where the direct object **John** is never marked, in Romanian the corresponding accusative proper name cannot be used without the marker **pe**:

(2) (a) Mary loves **John**

(b) Maria îl iubește **pe Ion** / ***Ion**

Mary CL₁⁵ loves *pe* John₁ / * John₁

In (b) **Ion** is the head and **pe** is the marker. The entire construction is therefore of type *hd-OBLmark-ph*.

In (b), the phrase **pe Ion** is coreferential with the pronominal clitic **îl** (“him”). The phenomenon is known under the name of ‘accusative clitic doubling’, and

⁵ Here ‘CL’ means clitic

independent facts (which will not be mentioned here) show that the head-marker phrase *modifies* the pronominal clitic. Obviously, what is needed is a generalization, according to which if something is a NP marked by **pe**, then that kind of linguistic object modifies the pronominal clitic.

The first idea in this respect would be to propose a constraint similar to C(*ph*). Nevertheless, this would not be a correct step because, unlike the generalization expressed by C(*ph*), the intended generalization would conflict with an important category of facts. It is about examples of head-marker phrases which are disallowed to modify a pronominal clitic, like in examples below:

- (3) (a) Ion a întâlnit **pe cineva**. (b) * Ion l-a întâlnit **pe cineva**.

John has met *pe* somebody John CL_i-has met *pe* somebody_i
'John met somebody.'

- (4) (a) Ion primește **pe oricine** în casa lui. (b) * Ion îl primește **pe oricine** în casa lui.

John hosts *pe* everybody in his house John CL_i hosts *pe* everybody_i in his house
'John hosts everybody in his house.'

- (5) (a) Ion nu a învinovățit **pe nimeni**. (b) * Ion nu l-a învinovățit **pe nimeni**.

'John did not blame it on anybody.'

- (6) (a) **Pe cine** nu iubește Ion ? (b) * **Pe cine** nu-l iubește Ion ?

Pe Whom not loves John? *Pe* Whom_i not CL_i loves John?
'Whom does not John love?'

What has to be done under these conditions? The most natural solution would be to adopt a constraint which avoids the counter-examples in (3)(b)-(6)(b). In HPSG, such a restriction is called a defeasible (or a default) constraint. In the present case, the appropriate default constraint simply has to specify that it has to be applied to all head-marker phrases of type *hd-OBLmark-ph*, less the ones enumerated under (3)(b)-(6)(b) – which are the only counter-examples to the needed generalization. The constraint in question looks as follows:

C(*hd-OBLmark-ph*)

! *hd-OBLmark-ph* ⇒

$$\left[\text{HEAD} : \text{noun} \left[\text{MOD} : \text{aff} - \text{ss} \left[\text{HEAD} : \text{noun} [\text{CASE} : \text{acc}] \right] \right] \right]$$

$$\left[\text{IND} : \text{ref} \right]$$

The symbol of the default in *C(hd-OBLmark-ph)* is !. The constraint means: if no other more specific constraint intervenes, a *hd-OBLmark-ph* modifies a corresponding pronominal clitic in accusative. Now, *C(hd-OBLmark-ph)* has to be followed by four non-default constraints which describe head-OBLmarker phrases that do not modify accusative clitic pronouns. These are noun phrases marked with *pe* which never participate in doubling structures: *pe cineva* (“somebody”), *pe oricine* (“anybody”), *pe nimeni* (“nobody”) *pe cine* (“whom”).

It is obvious from these considerations that constraints may be of two kinds: plain and defeasible. Defeasible constraints represent the HPSG way to deal with exceptions to generalizations. Another consequence of working with constraints - once it is known that they are formulated on types - is that they are arranged in a hierarchy, because the types themselves are.

2. HPSG Components

The system of notation described above and the devices which may be used within this system can be applied in principle to any discipline of linguistics, be it morphology, phonology, syntax, lexicon, semantics or pragmatics. In fact, one may speak about an HPSG phonology, morphology etc, even if there are differences between the degrees of elaboration of each of these components.

2.1. The Lexicon

The lexicon in HPSG has been paid special attention. Unlike the G&B Theory, HPSG is defined by the commitment to the thesis of the strong lexicalism, according to which the word integrity is not a matter of syntax. Consequently, words in HPSG are considered to be involved in syntactic processes only after the specification of their morphological, semantic, pragmatic and even syntactic properties. There are also two further consequences of this stance:

- lexical representations in HPSG are particularly reach
- an impressive amount of phenomena currently considered syntactic in their nature are accounted for in HPSG on the basis of appropriate lexical representations.

These characteristics justify the terms ‘grammar lexically driven’ or ‘lexicalist grammar’, which are also applied to HPSG.

In what follows, we will briefly present the main features involved in lexical representations.

2.1.1.1. Local Syntactic Properties of Lexical Items

A lexical item has no phrasal constituency, a fact which is noted either through the attribute-value pair [DTRS:*nev*], or by means of the pair [LEX: +]. A lexical item bears syntactic, semantic and (possibly) pragmatic information encoded into the value of the attribute SYNSEM. This information may be local (LOC) or non-local (NLOC). Local information specifies the category of a lexical item (CAT), the content (CONT) and the context (CTX). The value of the attribute CAT supplies information about part of speech (HEAD), subcategorization properties (VAL(ence)), argument selection (ARG-ST) and marking (MKING).

The HEAD feature is an important one, given the fact that its values have also to do with phrasal projections (see below, section 2.2.2.1). Here is a part from an inventory of the HEAD features, as proposed in Pollard and Sag (1987): 59-67, for English. It has to be said that several of these features may count as HEAD features for other languages, as well:

- Types specifying parts of speech: *noun*, *verb*, *adj(ective)*, *adv(erb)*, *prep(osition)*, *det(erminer)*, *mark(er)* etc.
- Types specifying case (for nominals) or forms (for other parts of speech): *nom(inative)*, *acc(usative)* etc, *fin(ite)*, *n-fin(ite)*
- Types specifying given properties of parts of speech:

- (i) Modification of a synsem (to be discussed in section 2.2.2.3: [MOD: ...]);
- (ii) Specification of a synsem (like in the case of determiners or markers: [SPEC: ...]),
- (iii) Negative form ([NEG: +])
- (iv) Predicativity: [PRED: +]. For example, a predicative preposition (*prep*[PRED: +]) is a preposition bearing semantic content (like **on** in the phrase **on the highway**). On the contrary, a non-predicative preposition (*prep*[PRED: -]) is a preposition with no semantic content (like **on** in the phrase **to rely on**...).

The value of the VAL attribute is a matrix with the following configuration:

AVM6

$$\left[\begin{array}{l} VAL : val \left[\begin{array}{l} SUBJ : list \\ SPR : list \\ COMPS : list \end{array} \right] \end{array} \right]$$

SUBJ, SPR and COMPS mean “subject”, “specifier” and “complements”, respectively⁶. Their values are lists (possibly empty) of ‘canonical synsems’. By ‘canonical synsem’ it is meant the type which represents the value of the attribute SYNSEM. This subtype of synsem designates the cluster of syntactic and semantic information borne out by a word or a phrase⁷. The consequence is that a clitic or a gap (like traces in the G&B Theory) does not bear a canonical synsem, this fact being important for syntactic matters (see below, AVMs7-8).

SUBJ and SPR have as value either the empty list or the list with only one member. Thus, no lexical item is expected to select more than one subject or specifier.

The value of the attribute ARG-ST is also a list (possibly empty). The difference between valences and arguments surfaces thanks to the difference between canonical and non-canonical synsems. Consider in this respect the example of the verb **a vedea** (“to see”) in Romanian. This verb selects two arguments which may coincide with the subject and the direct object if they are canonical synsems. In this case, the representation of the argument structure and the valence is like in AVM7, while one of the phrase projections of this valence representation is the example in (7):

AVM7

$$\left[\begin{array}{l} PHON : < vede > \\ \\ SYNSEM / LOC / CAT : \left[\begin{array}{l} VAL : \left[\begin{array}{l} SUBJ : < |1| can - ss > \\ COMPS : < |2| can - ss > \end{array} \right] \\ ARG - ST : < |1|, |2| > \end{array} \right] \end{array} \right]$$

(7) Ion vede un film
‘John sees a film’

On the other hand, it is also possible that the second argument of the verb **a vedea** is a pronominal clitic, like in (8):

(8) Ion îl vede
John CL-him sees
‘John sees him’

⁶ In HPSG a distinction is made between subjects and specifiers. In Pollard and Sag (1994):358-362, the reader may find several arguments in favor of this distinction.

⁷ The idea that the valence properties of a lexical item mean the selection of syntactic and semantic information is justified by a certain principle of locality: if a lexical item selects a valence it does not impose to that valence a given phonological form or a given architecture of daughters.

Recall that clitics do not bear canonical synsems; their synsem is of type *aff(ixal)*. A non-canonical synsem is not allowed to play a constituent role in syntax, but one cannot deny that it is a verb argument. So, such a synsem will never appear on the list of valences, yet it is allowed to occur on the list of the arguments. In this case the following asymmetry appears between valences and arguments:

AVM8

$$\left[\begin{array}{l} VAL : \left[\begin{array}{l} SUBJ : < |I| > \\ COMPS : < \diamond > \end{array} \right] \\ ARG - ST : < |I|, aff - ss[HEAD : noun[CASE : acc]] > \end{array} \right]$$

In the above representation, \diamond means the empty list. AVM8 proves that in a language like Romanian, the arguments of a lexical head may be different from its valences.

Another difference between valences and arguments is that arguments are specific only to lexical entries, while valences characterize both lexical items and phrases. The order of arguments (which is an order of obliqueness) is used to deal with matters of binding.

2.1.2. Non-local Syntactic Properties of Lexical Items

The lexicalist orientation of HPSG also surfaces in the treatment of unbounded dependencies. These are syntactic structures in which the order of the constituents does not mirror structural relations between them. Prototypically, unbounded dependencies are topicalizations (as in (9)(a)), partial interrogations (as in (b)) or relative constructions (as in (c)). What is common to all these structures is that the leftmost phrase is in relation with a lower position, which is indicated by the symbol $_$.

(9) (a) Bagels $_i$, John always said that he likes $_i$.

(b) What $_i$ did John always say that he likes $_i$?

(c) The bagels $_i$ that $_i$ John always said he likes $_i$ are made in Jim's bakery.

The lexicalist commitment of HPSG with respect to unbounded dependencies lies in the fact that in the explanation of these structures the form of the lexical entry plays an important role. This role may be rendered apparent from the examination of the verb *to like* in (). We will be focusing on the topicalization in (9)(a)⁸.

⁸ *Mutatis mutandi*, the explanation of interrogative and relative constructions is similar.

As specified in section 2.1.1, the arguments of the verb **to like** are not bound to be canonical synslems. Consequently, one may leave unspecified the nature of these synslems like in the following representation:

AVM9

$$\left[\begin{array}{l} \text{HEAD} : \textit{verb} \\ \text{ARG} - \text{ST} : \langle \textit{ss}[\text{HEAD} : \textit{noun}], \textit{ss}[\text{HEAD} : \textit{noun}] \rangle \end{array} \right]$$

This representation is compatible with the extensions in AVM9'-9'':

AVM9'

$$\left[\begin{array}{l} \text{HEAD} : \textit{verb} \\ \text{ARG} - \text{ST} : \langle |1| \textit{can} - \textit{ss}[\text{HEAD} : \textit{noun}], |2| \textit{can} - \textit{ss}[\text{HEAD} : \textit{noun}] \rangle \\ \text{VAL} : \left[\begin{array}{l} \text{SUBJ} : |1| \\ \text{COMPS} : |2| \end{array} \right] \end{array} \right]$$

AVM9''

$$\left[\begin{array}{l} \text{HEAD} : \textit{verb} \\ \text{ARG} - \text{ST} : \left\langle |1| \textit{can} - \textit{ss}[\text{HEAD} : \textit{noun}], \textit{gap} - \textit{ss} \left[\begin{array}{l} \text{LOC} : |1| \\ \text{SLASH} : |1| \end{array} \right] \right\rangle \\ \text{VAL} : \left[\begin{array}{l} \text{SUBJ} : |1| \\ \text{COMPS} : \langle \rangle \end{array} \right] \end{array} \right]$$

AVM9' represents a feature structure where both arguments of the verb **to like** are canonical synslems. This means that the verb will use them as valences, as well. On the contrary, in AVM9'', the second argument is not allowed to be realized as a valence, because it is the synsem of a gap. A gap is a linguistic object which describes

missing information ([SLASH:/1/]⁹). Missing information is ‘collected’ by the verb itself from its gap argument thanks to the following constraint on lexical entries:

C(SLASH amalgamation)

The SLASH value of a lexical head is the union of the SLASH values of its arguments.

Thanks to C(SLASH amalgamation) the lexical entry described under AVM9⁹ will instantiate the value /1/ as its non-local value:

AVM10

$$\left[\begin{array}{l} \text{HEAD} : \text{verb} \\ \text{ARG} - \text{ST} : < \dots, \text{gap} - \text{ss} \left[\begin{array}{l} \text{LOC} : |1| \\ \text{SLASH} : |1| \end{array} \right] \\ \text{NON} - \text{LOC} / \text{SLASH} : |1| \end{array} \right]$$

AVM10 may be now considered the right ‘starting point’ for the realization of the unbounded dependency in (9)(a). More precisely, AVM10 is a necessary condition for the existence of topicalizations¹⁰.

2.1.3. Semantic Properties of Lexical Items

The value of the attribute CONT(ent) is the following feature structure¹¹:

AVM11

⁹ Thus SLASH is a non-local attribute. Other ‘standard’ non-local attributes are REL and QUE, which are involved in the realization of relative and interrogative dependencies, respectively. The value of SLASH is a local structure (*loc*). The value of REL is an index – the index which is also the index of the noun modified by a relative clause. Finally, the value of QUE is the content feature structure composed of index and restriction (see below, the next section, especially AVM11)

¹⁰ Of course, AVM9⁹ is also subject to C(SLASH amalgamation) but given the fact that the SLASH values of its arguments are empty values, the SLASH value of the lexical head itself shall be empty, too. A lexical representation as described under AVM9⁹ cannot therefore be the right lexical ‘anchor’ for a topicalization.

¹¹ Actually, AVM11 is a simplification of the content representation. The content may be quantificational and non-quantificational. Quantificational content, as the name itself suggests, deals with matters of quantification. We will not be pursuing them here. The reader is referred to chapter 8 from Pollard and Sag (1994) or to Pollard (1998).

$$\left[\text{CONT} : \text{cont} \left[\begin{array}{l} \text{IND} : \text{ref} \vee \text{expl} \\ \text{RESTR} : \text{restr} \end{array} \right] \right]$$

IND means index. In section 1.2. it has been specified that an index consists in information about person, number and (possibly) gender. This information may be referentially relevant (as in the case of the pronoun *he* in English), or merely grammatically relevant (like in the case of the expletive English pronoun *it*). In the former case the index is referential, while in the latter it is expletive.

RESTR, on the other hand, means restriction. A restriction functions like a predication in predicate calculus: it takes an index as an argument, just like a predicate predicates something about an individual variable. This is shown in the value of the attribute RESTR which is a feature structure containing the attribute REL(atio)N and a corresponding number of semantic roles (SEM-ROLE). The value of RELN is a semantic type indicating the content of the lexical item. For instance, if the word is *cat* then the semantic type is just *cat*. And if the word is *see*, the semantic type is *see*.

As for semantic roles, they are similar to the thematic ones in the G&B Theory, with the difference that in HPSG they play no syntactic role. Semantic roles in HPSG are structurally connected to semantic types and describe the 'actors' associated with a given type. In the case of the word *cat*, for example, the semantic role following from the property of being cat is the instantiation (INST) of a cat by a given individual. The individual in question is referred to by means of the index of the word *cat*:

AVM12

$$\left[\begin{array}{c} \cdot \\ \text{CONT} : \left[\begin{array}{l} \text{IND} : |1| \text{ref} \left[\begin{array}{l} \text{PERS} : 3rd \\ \text{NB} : \text{sin } g \end{array} \right] \\ \text{RESTR} : \left[\begin{array}{l} \text{RELN} : \text{cat} \\ \text{SEM} - \text{ROLE} : \text{sem} - \text{role} [\text{INST} : |1|] \end{array} \right] \end{array} \right] \end{array} \right]$$

In the case of the verb *see*, the semantic roles are two: the 'experiencer' (he who sees) and the 'patient' (the person or the thing seen). The values of these semantic roles have to be the indexes of the arguments recorded in the ARG-ST list of the verb:

AVM13

$$\left[\begin{array}{l} CAT / ARG - ST : < ss[CONT / IND : |1|ref] ss[CONT / IND : |2|ref >] \\ \\ CONT / RESTR : \left[\begin{array}{l} RELN : see \\ SEM - ROLE : \left[\begin{array}{l} EXPERIENCER : |1| \\ PATIENT : |2| \end{array} \right] \end{array} \right] \end{array} \right]$$

Semantic types like *cat* or *see* are subtypes of more general types which are called 'nominal object' (*nom-obj*) and 'parametric state of affairs' (*psoas*)¹², respectively. In the case of nominal objects, an even more elaborated partition is put forward: nominal objects may be pronominal (*pron*) and non-pronominal (*npron*). A pronominal object in turn may be an anaphor (*ana*) or a personal pronoun (*ppron*). Finally, an anaphor may be reflexive (*refl*) or reciprocal (*rec*). Along with the argument order, this partition serves to building up a non-configurational binding theory.

2.1.4. Morphological Properties of Lexical Items

HPSG morphology deals with morphemic structures of words. According to the general format of the theory, morphology in HPSG means a set of constraints which rule morpheme combination.

The most general morphological attribute is MORPH. Its value is a feature structure composed of attributes ROOT and AFF(ix):

AVM14

$$\left[\begin{array}{l} MORPH : morph \left[\begin{array}{l} ROOT : root \\ AFF : pref \vee suff \end{array} \right] \end{array} \right]$$

Pref and *suff* mean prefix and suffix, respectively.

Morphology is the least elaborated part of HPSG, but a proper elaboration does not seem to raise serious problems. Among the main achievements, it is worth mentioning the distinction proposed in Sag and Miller (1997) between clitic and plain

¹² Parametric state of affairs is a concept coming from Situation Semantics (Barwise and Perry (1983)). As a matter of fact, the entire HPSG semantics is cast in the language of this semantic theory (see for example Pollard and Sag (1987)). Situation semantics however does not represent the monopoly of semantic representations in HPSG. For different orientations, see Lappin and Pollard (1999)

words, where the former ones are lexical items with ‘extended morphology’, that is, words incorporating affix-like elements¹³.

2.1.5. Pragmatic Properties of Lexical Items

As is well-known, certain words have ‘pragmatic’ properties which may consist in a special sensitivity to the utterance conditions. It is the case of the first person pronoun **I**, which uniquely denotes the person who speaks.

In HPSG, the most general feature dealing with pragmatic properties is C(on)T(e)XT. The value of this attribute is a feature structure formed of a special kind of indices (called ‘contextual’) and a background (BACKGRD). The value of the latter is a parametric state of affairs which represents a presupposition involved in the felicity conditions of an utterance. For instance, in the case of an utterance in which the speaker uses the pronoun **I**, one of the presuppositions that gives pragmatic meaning to the utterance regards the pronoun **I** and requires that this pronoun denote the speaker:

AVM15

$$\left[\begin{array}{l} \text{CONT / IND : } |1|_{\text{ref}} \left[\begin{array}{l} \text{PERS : 1st} \\ \text{NB : sg} \end{array} \right] \\ \text{CTXT : } \left[\begin{array}{l} \text{CTXT - IND : } \left[\begin{array}{l} \text{SPEAKER : } |1| \\ \dots \end{array} \right] \\ \text{BACKGRD : } \left[\begin{array}{l} \text{RELN : na min g} \\ \text{BEARER : } |1| \\ \text{NAME : I} \end{array} \right] \end{array} \right] \end{array} \right]$$

What this AVM says is that a presupposition of an utterance in which the pronoun **I** is used is the fact that the person denoted by the pronoun **I** is at the same time the speaker’s utterance. AVM15 characterizes the pragmatic property of the personal pronoun **I**.

The elaboration of the pragmatic component in HPSG absorbed particular energy and efforts in the past six years. In this respect one may quote the pioneering research of Elisabeth Engdahl and Enric Vallduvi on informational structure (Engdahl and Vallduvi (1995)). On the other hand, Georgia Green’s works substantively contributed to a comprehensive representation of pragmatics concepts in HPSG (Green (1997)).

¹³ For instance, weak pronouns in Romanian (**îl văd**), Italian (**lo vedo**) or French (**je le vois** - “I see him”)

2.1.6. Lexical Rules

In earlier versions of HPSG, lexical rules were designed to express morphological generalizations. Here is such a rule proposed for English in Pollard and Sag (1987): 208. The rule connects verbal base forms to third person (non-past) singular verbs:

Third-Singular LR

$$base \left[\begin{array}{l} PHON : |1| \\ SYNSEM : \left[\begin{array}{l} VAL : |2| \\ CONT : |3| \end{array} \right] \end{array} \right] \rightarrow 3rd\ sin\ g \left[\begin{array}{l} PHON : f3rdSING(|1|) \\ SYNSEM : \left[\begin{array}{l} VAL : |2| \\ CONT : |3| \end{array} \right] \end{array} \right]$$

The rule has an input and an output. The input is the type *base* which occupies a certain position in the hierarchy of lexical types for English. The output is a new lexical type, conventionally symbolized *3rd sing*. The rule modifies the phonology of the base form to the effect that a new phonology is obtained. According to this operation the base form **require** becomes **requires**, **work** becomes **works** and so on.

Late developments credit lexical rules with much more ‘abilities’. Thus, phenomena currently believed to be syntactic in their nature turn out to be well-suited with a lexicalist treatment in which lexical rules play a prominent role. Passive participle, for instance, is obtained by a lexical rule from the corresponding transitive verbs. Apart from changes in the phonology, the rule means a reorganization of the arguments of transitive verbs, along with the modification of the HEAD features:

Passive LR

$$st-v \left[\begin{array}{l} HEAD : verb \\ ARG-ST : \langle |1|, |2| \rangle \end{array} \right] \rightarrow pass-prpt \left[\begin{array}{l} HEAD : verb[VFORM : pass-prpt] \\ ARG-ST : \langle |2|, PP \left[\begin{array}{l} HEAD : prep[PFORM : by] \\ COMPS : |1| \end{array} \right] \rangle \end{array} \right]$$

The reorganization of the arguments means that the subject argument of the strict transitive verb (*st-v*) becomes the complement of the PP argument in the passive participle form, while the complement argument in the input turns into the subject argument in the output form. From this point of view, passivization amounts to a change in the obliqueness order of the arguments.

Lexical rules are subject to a long standing debate, because of their very nature: they are procedures, whereas the rest of the grammar is declarative (constraints are declarations, not operations). Some approaches attempted to remove them from the format of the grammar but the enterprise does not seem to enjoy wide acceptance.

2.2. Phrases

2.2.1. Phrase Typing

The first major achievement in the researches devoted to phrases is the hierarchy of the phrasal types. Phrases may be classified according to two criteria: clausality and headedness¹⁴. From the point of view of the head, phrases may be headed and non-headed (*hd-ph*, *nhd-ph*)¹⁵. Headed phrases bear a head and divide further into head-adjunct phrases and head-nexus phrases (*hd-adj-ph*, *hd-nex-ph*). Both the head constituent and the non-head one are daughters, and, more precisely, *signs*. For instance, in the case of the type *hd-adj-ph*, one has to have a head-daughter and an adjunct-daughter (for semantic properties of adjuncts and head-adjunct-phrases, see below, section 2.2.2.3.)¹⁶

The class of *hd-nex-phs* is formed of head-filler phrases, head-marker phrases and head-valence phrases (*hd-fill-ph*, *hd-mark-ph*, *hd-val-ph*). A filler-daughter is a constituent which ‘closes’ a long distance dependency. To resume the examples (9)(a)-(c) in section 2.1.2, filler-daughters are **bagels** (in (a)), the interrogative pronoun **what** (in(b)) and the relative pronoun **that** in (c). Head-daughters are the remaining phrases.

Head-marker phrases, on the other hand, are headed phrases where the non-head daughter is a marker. We already illustrated the idea of marker with the accusative NPs in Romanian (see above, section 1.6.). One may now add that in Romanian, a construction of type **pe Ion** is in fact a head-marker phrase, with the marker-daughter **pe**.

Finally, head-valence phrases divide into head-subject phrases, head-specifier phrases and head-complement phrases (*hd-subj-ph*, *hd-spr-ph*, *hd-comp-ph*). Each of these subtypes represents a realization of the corresponding valence encoded in the subcategorization frame (that is, the valence matrix) of the lexical item. For instance, if in Romanian, the verb **a iubi** (“to love”) selects a complement and a subject, these two valences have to be realized on the phrasal level, as, for instance, the phrases **iubește florile** (“... loves flowers”) and **Ion iubește florile** (“John loves flowers”). The former one is a head-complement phrase, whereas the latter is a head-subject phrase (with the

¹⁴ A classification according to the criterion of clausality is proposed in Ginzburg and Sag (2000). We refer the reader to that work.

¹⁵ Samples of non-headed phrases are constructions of coordination or free relative clauses.

¹⁶ Adjuncts are phrasal (for instance, relative clauses) or lexical constituents (for example, the majority of adverbs and adjectives) which modify a head. The distinction between arguments and adjuncts is difficult to make, and in fact a firm borderline between them cannot be drawn. Nevertheless, adjuncts are not generally considered arguments. They are rather mentioned on a separate list, the one of dependents (see Bouma, Malouf and Sag 1999). Dependents include both arguments and adjuncts and in the latest developments of HPSG, they serve as a basis for a unified treatment of extracted elements.

head-daughter **iubește florile**). An example of head-specifier phrase is the construction **a man** where **man** is the head and the determiner **a** is the specifier.

A specification is here necessary: this phrase hierarchy (proposed by Pollard and Sag for English) may be adapted to phrase descriptions in other languages, as well, but it does not claim at all to be a universal partition. More detailed analyses may lead to refinements and enrichments of this classification.

2.2.2. Constraints

2.2.2.1. Constraints Regarding Syntactic Local Features

A natural requirement concerning phrases is that they must satisfy at most one phrasal type. In other words, they cannot be at the same time (say) a head-complement phrase and a head-filler phrase.

Another general constraint is the HEAD Feature Principle (HFP). HFP is a constraint on headed phrases and regards, as its name suggests, the HEAD features. It stipulates that the HEAD features of a phrase have to be identical to those of its head-daughter:

HFP

$hd-ph \Rightarrow$

$$\left[\begin{array}{l} SS / CAT / HEAD : |1| \\ DTRS / HD - DTR : sign[SS / CAT / HEAD : |1|] \end{array} \right]$$

As it may be remarked, HFP rules the very existence of the headed phrases. In the absence of it, the idea of headed phrase would make no sense.

The Valence Principle (VALP) also acts on headed phrases but deals with valence realizations. It is a defeasible principle which says that if no other more specific constraint intervenes, in a headed phrase the valence value of the phrase is identical to the one of its head-daughter:

VALP

$!hd-ph \Rightarrow$

$$\left[\begin{array}{l} SS / CAT / VAL : |1| \\ HD - DTR : sign[SS / CAT / VAL : |1|] \end{array} \right]$$

Exceptions to this principle are exactly the subtypes of the type *hd-val-ph*: *hd-subj-ph*, *hd-spr-ph*, and *hd-comp-ph*. Indeed, in each of these cases, the appropriate VAL value of the phrase is distinct from the one of its head-daughter. Here is the constraint on the type *hd-subj-ph*:

$C(hd-subj-ph)$

$hd-subj-ph \Rightarrow$

$$\left[\begin{array}{l} SS / CAT / VAL / SUBJ : ev \\ HD - DTR : sign[SS / CAT / VAL / SUBJ : |1|can - ss] \\ SUBJ - DTR : sign[SS : |1|] \end{array} \right]$$

This constraint clearly shows that while the phrase as a whole has no availability to realize the subject (just because it already has a subject-daughter), the head-daughter displays the non-empty SUBJ value /1/, that is, a synsem which is part of the sign realized as the subject-daughter of the phrase. Put more simply, in the Romanian phrase

(10) Ion iubește florile.
“John loves flowers”.

there is no room for the subject realization because the subject is already realized. Yet the phrase:

(11)... iubește florile
“... loves flowers”.

which is the head-daughter of the phrase () and which is at the same time a head-complement phrase specifies that the value of its SUBJ attribute is non-empty.

The last constraint ruling local features which is discussed in this presentation is the Marking Principle (MKINGP). It rules the behavior of the feature MKING in a headed phrase and it stipulates that if no other more specific constraint intervenes, the MKING value of a headed phrase is identical to that of its head-daughter:

MKINGP

$!hd-ph \Rightarrow$

$$\left[\begin{array}{l} SS / LOC / MKING : |1| \\ DTRS : [HD - DTR : sign[SS / LOC / MKING : |1|]] \end{array} \right]$$

The exception is the plain constraint on the type *hd-mark-ph*, which requires that the MKING value of the phrase be coindexed with the MKING value of the marker-daughter:

$C(hd\text{-}mark\text{-}ph)$

$hd\text{-}mark\text{-}ph \Rightarrow$

$$\left[\begin{array}{l} SS / LOC / MKING : |1| \\ DTRS : \left[\begin{array}{l} HD - DTR : sign \\ MARK - DTR : sign[SS / LOC / MKING : |1|] \end{array} \right] \end{array} \right]$$

To come again to the example of the phrase **pe Ion** in Romanian (acc John), the phrase as a whole displays the pair [MKING:pe], because of the fact that it is of type *hd-mark-ph* and it must comply with the constraint $C(hd\text{-}mark\text{-}ph)$. On the contrary, in the phrase **Ion doarme** (“John is sleeping”), the MKING value of the whole phrase is coindexed with the MKING value of the head-daughter. The latter one is empty (because it is different from a head-marker phrase), and so it will be the MKING value of the phrase itself.

2.2.2.2. Constraints on Syntactic Non-local Features

In section 2.2.1 it has been said that a filler ‘closes’ a long distance dependency. To be now more precise, we have to say that the filler adds what the head-daughter is in need of. This idea is encoded in the following constraint on head-filler phrases:

$C(hd\text{-}fill\text{-}ph)$

$hd\text{-}fill\text{-}ph \Rightarrow$

$$\left[\begin{array}{l} SS / NLOC / SLASH : ev \\ DTRS : \left[\begin{array}{l} HD - DTR : sign[SS / NLOC / SLASH : |1|_{loc}] \\ FILL - DTR : sign[SS / LOC : |1|] \end{array} \right] \end{array} \right]$$

$C(hd\text{-}fill\text{-}ph)$ says that a head-filler phrase has to display empty value for the non-local feature SLASH. This is because the filler supplies the local information

which is indicated to be missing on the head-daughter (see the non-empty value of SLASH in $C(hd-fill-ph)$), and in doing so it ‘closes’ the dependence. ‘Closure’ consists in the fact that the local information of the filler is identical to the missing one displayed by the head.

In fact, the constraint is the reverse of the phenomenon described by the gap representation. Recall that a gap-synsem shows that its local information is exactly the absent information in a structure (see above, section 2.1.2, AVM9’). Further on, by the constraint on the amalgamation of SLASH values, the non-local information of the argument is shared with the one of its head. Finally, by $C(hd-fill-ph)$, ‘the order is re-established’: what is declared as missing reappears in the local structure of the filler.

There still remains to be explained how it is that the non-local information present on the head-daughter in $C(hd-fill-ph)$ is received from the lexical head which collects the same information through $C(SLASH\ amalgamation)$. Indeed, a link is missing here and it is supplied by a new constraint operating on headed phrases. The constraint is called ‘SLASH Inheritance Principle’ (SLIP) and it specifies that, in a headed phrase, the SLASH value of the phrase is identical to the SLASH value of its head-daughter.

Obviously, SLIP does what is needed. For example, in the case of sentence (i)(a), it forces the non-empty SLASH value of the word *like* to be shared with the phrase *he likes* ₁. The phenomenon repeats several times: by SLIP, the phrases

- (i) *that he likes* ₁
- (ii) *said that he likes* ₁
- (iii) *always said that he likes* ₁
- (iv) *John always said that he likes* ₁

all share the same non-local value. Thus, this value becomes accessible to the constraint $C(hd-fill-ph)$, which justifies the construction:

- (v) *Bagels* ₁, *John always said that he likes* ₁

One may now conclude that gap-synsems, the principle of the SLASH amalgamation, SLIP and the constraint on head-filler phrases are the HPSG devices designed to account for long distance dependencies. It may be remarked that the account does not involve the resort to movement.

2.2.2.3. Constraints on Semantic Features

As far as semantic (non-quantificational) properties of the phrases are concerned, they are ruled by the Semantic Principle (SEMP). This constraint has several versions. The

one presented here comes from Pollard and Sag (1994), and it is not the latest one¹⁷. The principle is a defeasible one and specifies that, by default, in a headed phrase the CONT value of the phrase has to be coindexed with the CONT value of its head-daughter.

SEMP

!hd-ph⇒

$$\left[\begin{array}{l} SS / LOC / CONT : |1| \\ DTRS / HD - DTR : sign[SS / LOC / CONT : |1|] \end{array} \right]$$

The exception is the constraint on the type *hd-adj-ph*. In this phrase, the CONT value of the phrase is identical to the CONT value of the adjunct-daughter. This accounts for the fact that, from a semantic point of view, an adjunct is the semantic head, while the syntactic head is the semantic argument of the adjunct:

C(hd-adj-ph)

hd-adj-ph⇒

$$\left[\begin{array}{l} SS / LOC / CONT : |1| \\ DTRS : \left[\begin{array}{l} HD - DTR : sign[SS : |2|] \\ ADJ - DTR : sign \left[SS / LOC \left[\begin{array}{l} CAT / HEAD : [MOD : |2|] \\ CONT : |1| \end{array} \right] \right] \end{array} \right] \end{array} \right]$$

2.2.2.4. Constituent Order Constraints

In HPSG, linear order is not considered a consequence of the structural relations between constituents. Principles of constituent order are therefore independent of matters of immediate dominance. The only link between the two is the fact that principles of linearization are formulated with respect to given phrasal types.

In earlier versions of HPSG, the study of linear order within the phrase led to certain language-dependent generalizations. English, for instance, was found to obey the following principles (Pollard and Sag (1987), chapter 7):

¹⁷ The latest version, to our knowledge, may be found in Sag and Wasow (1999). In this work there are in fact two principles: one accounting for semantic compositionality and another one accounting for inheritance of the semantic information.

C(CO)1

Lexical heads precede their complements.

C(CO)2

Non-lexical heads are preceded by their complements or functional words.

C(CO)3

Complements (other than clausal and infinitival) follow one another according to their obliqueness rank: a less oblique complement precedes a more oblique one.

These generalizations stimulated researches on other languages, as well. For French, Abeillé and Godard (1999) have shown that linear order correlates with a certain 'weight' property of constituents: constituents may be 'lite' or 'non-lite', the effect for linear order being that what is 'lite' is strictly adjacent to the head. For instance, in the sentence:

(12) La course donne soif à Jean

soif is bound to precede *à Jean*, because the former is lite, the latter is non-lite and the generalization is that a lite constituent precedes a non-lite one.

Other important contributions are researches of Michael Reape (Reape (1994)) and Andreas Kathol (Kathol (2000)). Kathol connects constituents with given domains, where a domain is the association between a phonological representation and a 'topological field' (a notion similar to the one used in traditional German grammar). Thus, one obtains models of constituent linearization within finite clauses or NPs, according to the number and the order of topological fields found in the language subject to investigation.

3. Final Remarks

What has been presented so far represents, so to speak, the 'hard core' of the theory. However, the 'HPSG world' is more complex. A category of researches is oriented towards the formal foundations of HPSG: it concerns studies in the logic of feature structures (King (1989), Carpenter (1992)). Other studies are focused on computational applications of the linguistic descriptions. Also, as a linguistic theory, HPSG puts forward a number of impressive challenges: it promotes a sound empiricism and rejects uncritically accepted theoretical entities. It shows that the concept of head possesses outstanding explanatory virtues. It pays attention to a wide range of linguistic data. It does not exaggerate the importance of syntax in the general economy of a linguistic theory. It is designed to offer itself to the main form of testability known in present-day

linguistics - the computational testability. Finally, it yields well-tempered hypotheses concerning the form of the Universal Grammar and the process of language acquisition (Green (2000)). All this makes HPSG a reliable tool for explaining, analyzing and processing facts of natural language.

ARGUMENTS, VALENCES AND DEPENDENTS. An HPSG APPROACH TO ACCUSATIVE CLITIC DOUBLING IN ROMANIAN

Verginica Barbu and Emil Ionescu

Abstract

This paper contributes the study of clitic doubling by concentrating upon accusative clitic doubling in Romanian. The analysis is cast in a non-derivational framework, namely Head-driven Phrase Structure Grammar (HPSG).

The main section is devoted to doubled NPs. It is argued that they display both complement and adjunct properties, and it is shown that, while the complement properties emerges from the examination of the relation between doubled NPs and the transitive verb, the adjunct ones are apparent if the relationship between doubled NPs and doubling clitics is taken into account. At the same time, the analysis emphasizes the versatile behavior of weak pronouns, which outside contexts of doubling are verb arguments but which in doubling structures loose this status.

Thanks to the flexible frame offered by HPSG, the approach easily captures the dual nature of doubled NPs. At the same time, due to the HPSG distinction between valences, arguments and dependents, it is explained how it is that while not loosing their referential properties, weak pronouns are not able to be arguments in structures of doubling. The analysis ends up with a natural generalization concerning the representation of the verbal transitivity in Romanian. Representations we propose are lexical. The analysis identifies three types of lexical entries responsible for all the varieties of verbal transitivity in Romanian: (non-pronominal) transitive verbs which select a direct object NPs just like in any other non-clitic language; (pronominal) transitive verbs, the direct object of which is clitically realized. And finally, pronominal transitive verbs which incorporate the clitic as a depedent but not as an argument. It is this last lexical representation which accounts for clitic doubling structures in a straightforward and natural manner.

1. Introduction

1.1. Theoretical Background

In the past five years, many HPSG works have provided massive evidence according to which what is currently termed 'adjunct' may be also taken as a complement¹⁸ (in this respect, see various types of data quoted in Bouma, Malouf and Sag (1999) p. 37-39).

¹⁸ 'Adjunct' and 'complement' are here taken in a non-configurational sense; that is, they are not positions but rather 'functions'.

On the other hand, recent theoretical HPSG developments have come to the threefold distinction between valences (VAL), arguments (ARG-ST) and dependents (DEPS - Bouma, Malouf and Sag (1999)). VAL is designed to connect the subcategorization frames of the lexical entries with corresponding phrase projections; ARG-ST deals with matters of binding, while the feature DEPS is designed to pave the way for a uniform treatment of adjunct and valence extraction.

The philosophy of 'adjuncts as complements' and the distinction between valences, arguments and dependents will be the theory backbone of the present analysis.

1.2. Clitic Doubling in Romanian and A Brief Survey of Its Generative Analyses

Romanian has in common with some Spanish dialects and several Balkan languages a structure currently known as 'clitic doubling' (CL-DB). Pronominal CL-DB in Romanian may be realized with accusative clitics (ACC-CLs), with dative ones or with both. We will be here concerned with accusative clitic doubling (ACC-CL-DB). A sample is presented in (1) below:

(1) Nu-l_i cunosc *pe* Ion_i
 Not CL-him_i know *pe* Ion_i
 'I don't know John'

In (1), the ACC-CL is *-l* ("him"), and the direct object (DO) is *Ion*. The DO is preceded by what is generally called in Romanian linguistics the preposition *pe*. Both the preposition and the clitic are obligatory in (1).

Examples like (1) represent a challenge for any grammatical theory positing that a given argument has only one realization. Thus, (1) seems to be a recalcitrant example, because what appears to be the direct object of the verb *a cunoaște* ("to know") is realized both as a weak pronoun and a full NP. So, at the first sight, either (1) has to be declared ill-formed, or the constraint on the argument realization should be abandoned (or modified). Nevertheless, as is well-known, neither solution is satisfactory, and, in fact, the commonplace strategy consists in the attempt to show that CL-DB is merely an apparent counter-example. This will also be the strategy of the present approach.

There is already a reach modern literature – a generative one - devoted to clitic doubling in Romanian. Here are what might be called its main theses:

◆ *Clitic Placement*

Hypothesis 1

The clitic is generated in the same position like any other NP complement of the verb. It moves to IP, which is assumed to be a Spec-less projection (Dobrovie-Sorin (1994) p. 54).

Hypothesis 2

The clitic is generated as part of the verb. The verb in turn is considered a lexical complex (Borer (1984), for clitics in general, and Cornilescu (1987), for pronominal clitics in Romanian).

♦ *The Status of the Clitic*

The clitic is a case absorber. Nevertheless, since it is not assigned to a structural position it cannot be the argument of the verb. The verb assigns thematic role only to the full NP (Jaeggli (1986) p. 17-18, for clitics in general, and Cornilescu (1987) p. 214, for clitics in Romanian).

♦ *The Sequence $cl_i \dots NP_i$*

A doubling clitic and a doubled NP form a discontinuous constituent (Cornilescu (1987) p. 215). The clitic is the head, while the doubled NP occupies the Spec position. (Gierling (1998) p. 79).

♦ *Case Assignment*

Hypothesis 1

Doubled NPs get case as a consequence of the fact that they are preceded by the preposition **pe**. (Cornilescu (1987), Dobrovie-Sorin (1994)).

Hypothesis 2

Doubled NPs get case under agreement, in a Spec-head configuration (Gierling (1998) p. 79).

♦ *Semantics*

Doubled NPs do not show quantificational properties. Doubled NPs are referential (Dobrovie-Sorin (1994) p. 234-235).

There are insightful remarks here but one cannot agree about all. For instance, it is not clear how one may reconcile the statement about the non-argument status of the clitic with its behavior in structures which are not of doubling (see below, section 4.1.1). Another problem: if it is true that doubled NPs are moved in the Spec position of the Clitic Phrase (as Gierling assumes in her analysis), how high has the clitic to move further, in order to leave behind the NP and to justify the order $cl_i \dots NP_i$? And, finally, what are the reasons to consider **pe** a preposition?

In what follows, we propose an HPSG analysis which naturally avoids all these problematic aspects. It may be summarized in the following three claims:

- In a structure of doubling, the verb incorporates the clitic as part of its extended morphology.

- In a structure of doubling, the clitic is neither an argument nor a valence of the verb. It is just a verb *dependent*. The only fact showing this dependence is the case: the verb assigns accusative to the clitic.
- In a structure of doubling, the clitic-doubled full NP leads a double life: it behaves like a complement in its relation with the verb and at the same time it behaves like an adjunct in its relation with the clitic.

The structure of the paper is the following: in section 2, we give a description of ACC-CL-DB in Romanian. In section 3, we identify the issues which lead to an appropriate analysis of the phenomenon. Section 4 is devoted to an analysis of doubled NP, while section 5 presents the clitic doubling phenomenon in the perspective of the HPSG concepts mentioned in the previous section.

2. Accusative Clitic Doubling in Romanian: Description

ACC-CL-DB in Romanian has the following properties¹⁹:

1. The (transitive) verb is accompanied by one of the weak pronouns from the paradigm given below (the pronouns are personal)²⁰:

	SINGULAR	PLURAL
FIRST PERSON	mă, m-	ne
SECOND PERSON	te	vă, v-
THIRD PERSON	îl, l(-) (masc.); o (fem.)	îi, i(-) (masc.); le (fem.)

2. The NP which is clitic-doubled occupies the post-verbal position, and is obligatorily preceded by the so-called preposition *pe*²¹.

(2) Nu-l_i cunosc *(pe) el_i
 Not CL-him_i know *pe* him_i
 'I don't know him'

¹⁹ Clitics placement and clitics linearization are not discussed in this paper (see Dobrovie-Sorin (1994) p. 49-81 for an analysis in the GB perspective, and Monachesi's HPSG account (Monachesi (2000)).

²⁰ Throughout this paper, the terms "weak pronoun" and "clitic" will be conventionally considered synonymous.

²¹ If the doubled NP is in preverbal position, the structure is not of doubling but of *clitic left dislocation*, e.g:

(i) Eu romanul_i l_i -am citit.
 I novel-the_i ACC CL-it_i -have read
 'The novel, I read.'

Comment:

Unlike Italian, in Romanian, the subject and the complement are not bound to be adjacent to the verb. Thus, structures in which the subject comes after the verb and is followed in turn by a clitic-doubled object also qualify in Romanian as *accusative clitic doubling*, while in Italian they are *clitic right dislocations*:

(3) Lo legge Gianni, il giornale.²²
ACC CL-it_i reads John the newspaper_i
'John is reading the newspaper.'

(4) O iubește Ion pe Ioana.
CL-her_i loves John *pe* Joanna_i
'John loves Joanna.'

3. Doubling consists in the fact that the weak pronoun and its full NP pair agree in case, person, number, and (for the third person) in gender.

4. The set of doubled NPs is a subclass of the set of accusative NPs. For instance, accusative strong forms of the personal pronoun have to be doubled:

(5) Nu-*(I_i) cunosc *pe* el_i
Not CL-him_i know *pe* him_i
'I don't know him.'

On the contrary, the quantifier *cineva* ('somebody') may not :

(6) Ion (* îl/o) urăște *pe* cineva.
John CL-him/her_i hates *pe* somebody_i
'John hates somebody.'

5. Clitic-doubled strong pronouns present the peculiarity of being obligatorily under focus. The unmarked way of indicating the referent of the DO is to use the corresponding weak pronoun alone:

(7) (a) Ion *mă* vizitează.
John CL-me visits
'John visits me.'

The focus version of (7) (a) is a structure of doubling. This focus in situ is expressed through intonation, appropriate determiners (such as *chiar* ('even'), *numai* ('only'), *doar* ('mere'), *și* ('too')), or both:

²² The example is borrowed from Sanfilippo (1997) p. 357.

(7) (b) Ion mă vizitează (numai/chiar/doar/și) PE MINE.

John CL-me (only/even/too) visits PE ME

‘John visits (only/even) me (too).’

In spite of the fact that this variety of focus is somewhat obscured by the relative free word order (which provides the standard option of expressing focus), it does exist: non-pronominal doubled NP may be focused optionally, while a ‘unmarked’ version of non-pronominal doubled NP is at least odd:

(8) Ion îi așteaptă pe părinții săi/PE PĂRINȚII SĂI. (9) Ion te așteaptă pe tine (??) /
PE TINE.

John CL-them waits *pe* parents his/PE PARENTS HIS John CL-you waits *pe*
you/PE YOU

‘John is waiting for his parents/for HIS PARENTS.’

3. Approaching the Issue

To give an explanation for the characteristics enumerated under 1-5, we need to answer the following questions:

- (i) What is the status of ACC-CLs ?
- (ii) What is the status of *pe* ?
- (iii) What is the status of doubled NPs ?

As for the first two questions we will rely on the conclusions of some previous analyses (Ionescu (1996), Barbu (1998), Monachesi (2000), Ionescu (2001)). The conclusions of these analyses are the following:

- ACC-CLs

- (i) ACC-CLs fail to pass constituency tests, such as coordination or interrogativization. Consequently, they cannot be considered lexical constituents and they cannot have any syntactic independency.
- (ii) Instead, ACC-CLs answer better tests for affixes. They manifest a high degree of selection with respect to their host; they exhibit arbitrary gaps in the set of combinations with the host, they are rigidly and idiosyncratically ordered, and they take narrow scope over coordination (Sag and Miller (1997)). For all these reasons, ACC-CLs in Romanian are here considered a part of the transitive verb.
- (iii) If it occurs in structures which are not of doubling, an ACC-CL is the direct object argument of the verb, while if it doubles a non-pronominal NP it becomes non-argumental.

• The ‘Preposition’ *pe*

- (i) *Pe* involved in the DO construction is neither a case assigner, nor a preposition.
- (ii) It marks NPs which independently receive accusative.
- (iii) *Pe* has an empty content (fact which does not have to be conflated with the one that *pe* marks only certain semantic types of accusative NPs).

The fact that *pe* marks accusative NPs is encoded in its lexical description. The fact that not every accusative NP may be marked with *pe* is also encoded in the lexical description of *pe*, thanks to a default device that specifies which NPs escape *pe*-marking. *Pe* will be here called ‘OBL(igueness) marker’. The phrase it builds along with the head NP is a subtype (specific to Romanian) of the phrase *hd-mark-ph* (for *hd-mark-ph* see Pollard and Sag (1994) p. 44-46) It is abbreviated here *hd-OBL mark-ph*.

4. The Status of Doubled NPs

In this section we will be showing that doubled NPs instantiate both properties of complements and adjuncts. Some uncontroversial complement features are: complements are arguments of a head and are listed as valences of that head; they are also semantic arguments and their occurrence is in general obligatory. Adjuncts, on the other hand, are semantic functors in relation to their syntactic heads and in general they are not valences of the head they modify.

4.1. Complement Properties of Doubled NPs

Evidence that doubled NPs display complement properties comes from binding, semantic role assignment and passivization. Indirect evidence that pronominal doubled NPs are also complements is supplied by structures of clitic left dislocation.

4.1.1. Semantic Role Assignment and Binding

Non-pronominal doubled NPs clearly show that they receive semantic role from the verb and that they are involved in binding relations. This pleads for their argument nature. As for binding, if a doubled NP is an R-expression, it shall be free throughout, as (10) shows²³:

- (10) Ion_i crede că poliția_j îl urmărește pe Petre_{i/*j/k}
‘John_i believes that police_j watches Peter_{i/*j/k}’

What has to be also noticed with respect to binding is the non-uniform behavior of ACC CLs. If a weak pronoun does not double a NP, it behaves according to its pronominal nature, that is, it is locally free:

²³ It is not relevant for the present discussion to specify the kind of Binding Theory one adheres to.

- (11) Ion_i crede că poliția_j îl_{i/*j_k} urmărește.
John_i believes that police-the_j CL-him_{i/*j_k} watches
‘John_i believes that police_j watches him_{i/*j_k}’

On the contrary, if the weak pronoun doubles an R-expression what counts for binding is the NP. For instance, in (10), written below as (12), the weak pronoun *îl* (“him”) cannot be bound to the NP *Ion*, just because *pe Petre* is an R-expression:

- (12) Ion_i crede că poliția_j îl_{i/*j_k} urmărește *pe Petre*_{i/*j_k}
John_i believes that police-the_j CL-him_{i/*j_k} watches *pe Peter*_{i/*j_k}
‘John_i believes that police_j watches Peter_{i/*j_k}’

We take this fact as proving that ACC-CLs which double R-expressions are non-argumental.

4.1.2. Clitic Left Dislocation

In the case of pronominal NPs, both binding and semantic role assignment fail to be diagnostic tests for the argument nature of doubled NPs, because pronominal NPs share the binding type with their copies. Under these conditions it is not possible any more to determine whether the contribution to binding is that of the weak pronoun or, on the contrary, that of the strong one. The same comes about with semantic role assignment. However, indirect evidence that pronominal NPs are also DOs is supplied by their behaviour with respect to clitic left dislocation. In (13) (a)–(b) what is dislocated left to the left is a strong pronoun and a referential expression, respectively. We already know that non-pronominal doubled NPs are arguments of the verb. Under these conditions, it is relevant that both non-pronominal and pronominal NPs behave identically in left dislocations, because this might mean that the strong pronoun has the same argument status:

- (13) (a) **Pe el**_i *îl*_i iubește _j Ioana. (strong pronoun dislocation).
 ‘It is him that Joanna loves’

- (b) **Pe Ion**_i *îl*_i iubește _j Ioana (non-pronominal DO dislocation)
 ‘It is John that Joanna loves’

4.1.3. Passivization

No matter what theory of passivization one adheres to, there is almost general agreement that passivization affects the direct object argument of the verb. If one relies on this minimal assumption, the trivial conclusion concerning structures of doubling is that a doubled NP is a verb complement and hence a verb argument, because in a passive structure what appears to be the subject of the passive form turns out to be the doubled NP in the corresponding active construction:

- (14) Poliția îl urmărește *pe* Ion. (15) Ion este urmărit de către poliție.
 ‘Police watches John.’ ‘John is watched by the police.’

4.2. Adjunct Properties of Doubled NPs

Relevant to the adjunct nature of doubled NPs are the failure of pronominalization, the way doubled NPs get case, the semantic relationship between a doubled NP and its pronominal copy, and the obligatory focus on the strong pronoun.

4.2.1. Focus

As noticed in section 3, strong pronouns which are doubled are subject to obligatory focus. This fact contrasts with the one that non-pronominal doubled NPs are *optionally* focus-marked. We know now that non-pronominal NPs are complements of the verb. Consequently, if regarded in this perspective, optional focus may be explained by the fact that the main role of non-pronominal NPs is that of being verb arguments. On the contrary, if focus is obligatory in the case of pronominal NPs, this might mean that pronominal NPs are not arguments of the verb – if they were, they would be also optionally focus-marked. So, we take the rather ill-formed structures like:

- (16) (??) Ion te așteaptă *pe* tine.
 John CL-you waits *pe* you
 ‘John is waiting for his parents’

(where the strong pronoun is not under focus) as a possible violation of the requirement that the verb selects only one argument.

4.2.2. Pronominalization

In Romanian, uncontroversial direct objects pass the pronominalization test, by systematically allowing substitution with corresponding ACC-CLs. In such a situation, a weak pronoun becomes the verb argument:

- (17) (a) Ion citește **o carte**. (b) Ion **o** citește.
 ‘John is reading a book.’ John CL-her/it is reading
 ‘John is reading it.’

Pronominalization, though, fails in the case of doubled NPs. If such a NP is substituted with an ACC-CL, the construction is ill-formed:

- (18) (a) Ion **o** iubește *pe* Ioana. (b) * Ion **o o** iubește.
 John CL-her loves *pe* Joanna John CL-her CL-her loves
 ‘John loves Joanna.’

It would be a mistake to believe that ungrammaticality might originate here in that the weak pronoun which replaces the doubled NP is a redundant argument of the verb (that is, it fulfills a function already fulfilled by the other weak pronoun). What is wrong with this hypothesis is the fact that, as binding shows, the weak pronoun *is not* an argument. It is only the doubled NP which shows argument properties.

By itself, the failure of pronominalization does not supply evidence that doubled NPs bear adjunct properties. It only points out that doubled NPs are not standard DOs - because standard DOs in Romanian, participate in binding and may be pronominalized. Nevertheless, what gives special relevance to the failure of pronominalization is the fact that the entire sequence weak pronoun-NP *may* be replaced with a weak pronoun:

(19) Ion **o** iubește **pe Ioana**.

(20) Ion **o** iubește.

The substitution of the sequence **o... pe Ioana** with the weak pronoun **o** in (19) is not vacuous pronominalization²⁴. And if corroborated with the failure of pronominalization in contexts such as (18) (b), the substitution in (19) leads to the following conclusions:

- (a) The sequence weak pronoun-NP functions as a kind of phrase, in which the weak pronoun appears to be the head and the NP seems to play the role of a non-head daughter.
- (b) Since, as a part of speech, the weak pronoun is a noun, the entire 'phrase' seems to be a NP, too.

If these conclusions are correct, the sequence weak pronoun-NP possibly instantiates either the phrase(-like) type (*nominal*) *head-specifier* or the type (*nominal*) *head-adjunct*. The next two sections will show why other types of nominal phrases (such as head-complement or head-subject) are excluded from investigation.

4.2.3. Case Marking

A commonplace claim in the literature devoted to ACC-CL-DB in Romanian, is that a doubled NP receives accusative from the preposition *pe* which marks the NP. However, there is no convincing evidence that in structures of doubling, *pe* is a preposition. On the contrary, there are good reasons to claim that this *pe* is a marker which specifies a NP, the case of which is received independently. So, the issue about how doubled NPs get accusative still remains open.

Under the hypothesis that *pe* is a marker not involved in case assignment, one cannot state that in a structure of doubling it is the verb which assigns accusative to the doubled NP. The verb only assigns accusative to the weak pronoun, fact which means that it is quite impossible for it to be a case source for doubled NPs, too. Instead, what is particularly significant is the case agreement between the weak pronoun and the doubled NP. This agreement actually indicates that doubled NPs do not receive accusative under verb's argument selection.

²⁴ Vacuous pronominalization in this case would mean the substitution in the sequence **o...pe Ioana** of a token of **o** with another token of **o**.

One cannot avoid to relate this fact to data of pronominalization noticed in the previous section. And if one makes this correlation, one gets new evidence that such a sequence instantiates either the type (nominal) head-specifier or (nominal) head-adjunct: indeed, it is in such structures that the non-head daughter receives case by agreement from its head. However, still no evidence is available for specifying which one of these two phrase types is involved in structures of clitic doubling.

4.2.4. The Content Relationship between Doubled NPs and Doubling CLs

It is known that a prominent peculiarity of adjuncts is the fact that they behave as semantic functors: they are semantic heads of the phrases they belong to, because they take the content of the syntactic head and make a new content.

This property of adjuncts is illustrated by the manner in which doubled NPs behave with respect to weak pronouns. Recall that a weak pronoun has no descriptive content. Nevertheless, the fact that it shares the same information of person, number and gender with the doubled NP, allows the full NP to modify the empty content of the pronoun. The relation is modification not specification, because the contribution of the doubled NP to the content of the NP it belongs to is not empty (as in the case of specifiers): if considered in its relation to the doubling clitic, a doubled NP carries new semantic information, that is, information which is not present in the content of the weak pronoun.

This adjunct behavior of doubled NPs can be made even more visible if looked from the formal semantics perspective. In a comprehensive work on several clitic doubling languages, Gutiérrez-Rexach (Gutiérrez-Rexach (1999)) argues that accusative clitics denote what in the theory of generalized quantifiers is called a *principal filter*. Although the author does not insist on the denotation of the entire sequence clitic-doubled NP, it is pretty clear that this is a principal filter, too. How then to interpret the denotation of the doubled NP? A plausible way to refer to it is to say that it is a function from a principal filter (the weak pronoun) to another principal filter (the sequence weak pronoun-NP). But to say this is to resort to the pattern of nominal modification which has been just invoked above.

5. Modeling Accusative Clitic Doubling in HPSG

5.1. Adjuncts as Complements

Evidence gathered so far shows that adjunct and complement properties of doubled NPs are equally important. Actually, it is this combination of features which defines the very nature of these constituents. Under these conditions it would be a mistake to force the conclusion that one set of properties prevails over the another one.

The dual nature of doubled NPs in Romanian represents in fact a new piece of evidence in favor of what might be called 'the philosophy of adjunct as a complement'²⁵. As mentioned in the introduction of this paper, this stance is specific to many HPSG works, and

²⁵ With the only difference that in our case the doubled NP shows properties of complement in relation to its governing verb and properties of adjunct in relation to the doubling clitic.

relies above all on evidence according to which what is currently termed ‘adjunct’ may be taken as a complement, too. In what follows, facts commented above will be modeled in accordance with this philosophy of adjunct as a complement.

To achieve this goal, two steps have to be done. First, we ought to give a constraint according to which doubled NPs modify the doubling weak pronoun. And then, we have to supply the form of the lexical types accounting for the main types of transitive constructions in Romanian²⁶.

5.2. Constraints on Doubled NPs

The needed constraint on doubled NPs has to encode the fact that they modify weak pronouns. Since doubled NPs are marked with the marker *pe*, the constraint has to be given on the phrase type head-OBL marker (*hd-OBL marker-ph*). On the other hand, since not every *hd-OBL marker-ph* modifies a verb with weak pronoun, the constraint must be a default one:

C1

$$/hd - OBLmark - ph \Rightarrow \left[HEAD : noun \left[MOD : aff - ss \left[\begin{array}{l} HEAD : noun [CASE : acc] \\ CONT | IND : ref \end{array} \right] \right] \right]$$

(In C1, / is the default symbol and \Rightarrow means implication). C1 has to be read as follows: “if no other more specific constraint intervenes, a head-OBL marker phrase has to modify the affix synsem indicated as the value of the feature MOD(ifies)”.

C1 is followed by four non-default constraints which describe head-OBL marker phrases that do not modify weak pronouns. These are noun phrases marked with *pe* which never participate in doubling structures. In other words, they are direct objects selected by verbs that do not incorporate weak pronouns.

The specification [MOD: *ev*] is part of the lexical representations of the items involved in those head-OBL marker-phrases which are exceptions to C1. This specification is transmitted to the phrase itself thanks to the Head Feature Principle. The lexical items in question are *cineva* (‘somebody’), *oricine* (‘everybody/anybody’), *cine* (‘who’) and *nimeni* (‘nobody’). Specification [MOD: *ev*] explains the contrasts below:

(21) (a) Ion a întâlnit **pe cineva**. (b) * Ion l-a întâlnit **pe cineva**.

John has met *pe* somebody John CL-him_i-has met *pe* somebody_i
‘John met somebody.’

(22) (a) Ion primește **pe oricine** în casa lui. (b) * Ion îl primește **pe oricine** în casa lui.

John hosts *pe* everybody in his house John CL-him_i hosts *pe* everybody_i in his house

²⁶ In spite of some explanations, this part of the paper assumes a certain familiarity of the reader with basics of HPSG.

‘John hosts everybody in his house.’

- (23) (a) Ion nu a învinovăţit **pe nimeni**. (b) * Ion nu l-a învinovăţit **pe nimeni**.
‘John did not blame it on anybody.’

- (24) (a) **Pe cine** nu iubeşte Ion ? (b) * **Pe cine** nu-l iubeşte Ion ?
Pe Whom not loves John? Pe Whom_i not CL-him_i loves John?
‘Whom does not John love?’

In (b) examples, ungrammaticality comes from the fact that a *pe*-marked NP which is specified as [MOD: *ev*] is the complement of a verb which requires a *pe*-marked NP whose MOD value has to be the one in C1.

5.3. Lexical Representations of Verbal Transitivity

The analysis developed above shows that transitive constructions in Romanian may be realized in three different ways:

- (i) A transitive verb may select a DO full NP:

- (25) Ion iubeşte **femeile**.
‘John loves women.’

- (ii) A transitive verb may have as argument a weak pronoun:

- (26) Ion l-a întâlnit.
‘John met him/it’

- (iii) A transitive verb may select a DO doubled by a weak pronoun which is not an argument²⁷:

- (27) Ion a ignorat-o **pe Ioana**.
‘John has ignored CL-her *pe* Joanna.’
‘John ignored Joanna.’

It may be that ‘the same verb’ satisfies each of these frames. For instance, the verb **a iubi** may take a full NP (Ion iubeşte **femeile** ‘John loves women.’), a clitic argument (Ion **le** iubeşte ‘John loves them.’) or a doubled NP (Ion **le** iubeşte **pe toate colegile de serviciu** ‘John loves all his office mates’ Ion **le** iubeşte **ŞI PE ELE** ‘John loves [_F them, too]).

²⁷ The first three types have been suggested to us by Ana-Maria Barbu.

Differences between these types of transitive verbs may be easily expressed by exploiting the features VAL, ARG-ST and DEPS (see Bouma, Malouf and Sag (1999)). The values of these features are lists of synsems²⁸. The relationship between them may be summarized in the following statements:

- Not every valence is an argument, and also not every argument is a valence: indeed, there may be expletive subjects and objects; on the other hand, an argument is allowed to bear a non-canonical synsem, while a valence always bears a canonical one²⁹).
- Every valence is also a dependent but the converse does not hold necessarily (because a dependent is also allowed to bear a non-canonical synsem, while the synsem of a valence is always canonical).
- Every argument is also a dependent but the converse does not hold necessarily (because it may be that a dependent plays no role in matters of argument structure).

With these specifications, the relevant parts of each lexical type identified previously look as follows:

(i') A transitive verb incorporating no clitic has to satisfy the conditions described in matrix M1:

M1

$$\left[\begin{array}{l} \text{HEAD} : \text{trans} - \text{verb} \\ \text{VAL} : \left[\begin{array}{l} \text{SUBJ} : |1| \\ \text{COMPS} : |2| \text{ss} [\text{HD} : \text{noun} [\text{MOD} : \text{ev}]] \end{array} \right] \\ \text{ARG} - \text{ST} : |1| \oplus |2| \\ \text{DEPS} : |1| \oplus |2| \end{array} \right]$$

As seen in M1, such a verb selects an NP complement which has to be specified as [MOD: ev]. What is a valence (that is, a subject or a complement) is also an argument and what is an argument is also a dependent. The complement is underspecified with respect to **pe**-marking, fact which is correct, because a complement of a 'non-clitic' transitive verb may be either **pe**-marked or unmarked.

²⁸ A *synsem* is the most important fragment of linguistic information in HPSG; It displays syntactic, semantic and pragmatic information.

²⁹ A *canonical synsem* is the synsem of a linguistic object which is allowed to participate in syntactic structures. For instance, the synsem of a word or of a phrase is canonical. A *non-canonical synsem*, on the other hand, is the synsem of a linguistic object which cannot be involved in syntactic constructions. Synsems of affixes or synsems of 'empty categories' are noncanonical.

(ii') A verb incorporating a weak pronoun with no doubling function must obey the constraint expressed in M2 (We dub verbs with incorporated pronoun 'clitic verbs' (*trans-cl-verb*))³⁰:

M2

$$\left[\begin{array}{l} \text{HEAD : } trans - cl - verb \\ \text{VAL : } \left[\begin{array}{l} \text{SUBJ : } |1| \\ \text{COMPS : } el \end{array} \right] \\ \text{ARG - ST : } |1| \oplus |2| \text{aff - ss} \left[\begin{array}{l} \text{HEAD : } noun [CASE : acc] \\ \text{CONT : } \left[\begin{array}{l} \text{IND : } |3| \text{ref} \\ \text{RESTR | RELN : } pron \end{array} \right] \end{array} \right] \\ \text{DEPS : } |1| \oplus |2| \end{array} \right]$$

In M2, the COMPS list is empty because the direct object argument (i.e. tag /2/) is realized as an affix, and an affix is not allowed to have phrasal projection. Instead, the affix appears both in the ARG-ST list and the DEPS one. This is correct: a non-canonical synsem may be a dependent or (like in the present case) both a dependent and an argument.

(iii') Finally, the lexical representation accounting for doubled DOs looks as follows:

M3:

$$\left[\begin{array}{l} \text{HEAD : } trans - cl - verb \\ \text{VAL : } \left[\begin{array}{l} \text{SUBJ : } |1| \\ \text{COMPS : } |2| \text{ss} \left[\begin{array}{l} \text{HEAD : } noun [MOD : |3| \text{aff - ss}] \\ \text{CONT | RESTR | RELN : } non - anaph \\ \text{MKING : } pe \end{array} \right] \end{array} \right] \\ \text{ARG - ST : } |1| \oplus |2| \\ \text{DEPS : } |1| \oplus |2| \oplus |3| \end{array} \right]$$

³⁰ CONT means 'content', IND means 'index' and it refers to the information of person number and gender of the noun. The symbol *ref* shows that the noun is not expletive. RESTR and RELN mean 'restriction' and 'relation, respectively. Finally, *pron* means 'pronominal'.

COMMENTS:

- Since arguments are here canonical synsems, what is an argument is also a valence. The complement bears the binding type *non-anaph(orical)*, fact which means that it may be a pronoun or an R-expression.
- It may be noticed that the list of dependents is richer. It comprises now a new dependent, in fact the weak pronoun itself. The weak pronoun does not appear on the valence list (because it bears non-canonical synsem), nor does it belong to the ARG-ST list (because it is not involved in binding). Nevertheless, it is a verb dependent, as the verb clearly selects it by assigning it accusative.
- According to M3, the valence /2/ modifies the synsem of the pronominal affix, yet no head-adjunct phrase is projected accordingly, because /2/ appears in the valence list. The only phrase projected as a consequence of the complement realization is of type head-complement.

5.4. Final Remarks

♦ It is apparent from the analysis developed above that transitivity in Romanian is a highly constrained phenomenon which is determined by numerous distinctions:

- (i) The distinction between verbs which impose to their DO the content type *non-animate* (or a subtype of) and verbs which do not impose this restriction.
- (ii) The distinction between lexical and clitic realization of the DO argument.
- (iii) The distinction between argument and non-argument realization of the weak pronoun.
- (iv) The distinction between marked and unmarked DOs.
- (v) The distinction between modifying and non-modifying NPs.

The order of (i)-(v) means more and more language-specific constraints. As for the first distinction, it looks like a language universal and accounts for the fact that verbs of the former category may only take as a pronominal DO pronouns of the third person:

(28) (a) Ion a îndeplinit misiunea/ordinul/*copilul.
I-a îndeplinit.

(b) Ion a îndeplinit-o./ Ion

John has accomplished FEM-task-the/MASC-order-the/*child-the. John has accomplished FEM CL-it/

John MASC CL-it has

accomplished
'John accomplished the task/the order/*the child'
accomplished it'

'John

(c) * Ion m-/te-a îndeplinit.

Pronominalization irrespective of the person of the pronoun is possible only with verbs which do not impose this restriction (see (29) (a)-(b)), and also with verbs which require that their DO be marked *animate* or *human* (see (30) (a)-(b)):

(29) (a) **Ion așteaptă un autobuz/un coleg.** (b) **Ion mă/îl așteaptă.**

John waits a bus/a colleague. **John CL-me/ CL-him waits**

‘John is waiting for a bus/colleague’ ‘John is waiting for

me/him’

(30) (a) **Poliția a arestat demonstrații/*autobuzul.** (b) **Poliția m/te/îl-a arestat.**

Police-the has arrested protesters-the/bus-the

Police-the CL-me/CL-you/CL-him

has arrested

‘The police arrested the protesters/*the bus.’

‘The police arrested

me/you/him.’

The second distinction explains that just like in French or Italian, Romanian makes use of clitic arguments. The third shows that, in addition, in Romanian clitics may be non-argument dependents, too. The fourth places Romanian in the same family with Spanish, where some DOs are also marked (with the ‘preposition’ *a*). Finally, the fifth distinction explains what a doubled NP is: it is a NP which modifies the weak pronoun. Within the HPSG framework, one may encode all these distinctions, thanks to a hierarchical system of lexical constraints with multiple inheritance.

♦ On the present view, the role of the weak pronoun is complex. Actually, one may talk about several *roles* of the weak pronoun: it may be the non-canonical argument of the verb, or (if it is modified by a full NP) it becomes in addition the case source for the doubled NP. The fact that it preserves its referential properties discards the hypothesis that accusative clitic doubling in Romanian is a form of agreement marking.

♦ The fact that the argument clitic appears on the list of dependents does not mean that it may be subject to extraction, because in a strong lexicalist framework like HPSG, affixes are not allowed to be extracted. We are thus able to also provide explanation for the fact that a structure like:

(31) *Îl_i iubește_i Ioana
Him_i loves_i Joanna

is ill-formed.

♦ We think that the pattern of explanation proposed in this paper may be applied to other clitic doubling languages or dialects, as well (for instance, Bulgarian or River Plate Spanish). In presuming this, we are referring to the relation of modification between doubled NPs and

doubling clitics. Indeed, it seems that modification is general enough to cover all cases of clitic doubling, irrespective of the reasons which determine the realization of this type of structure (reasons which, for sure, differ from language to language).

BIBLIOGRAPHY

- Abeillé, A. and D. Godard**, *French Word Order and Lexical Weight*, in **R. Borsley** (ed.), *Syntax and Semantics*, vol 30. *Syntactic Categories*, Academic Press, New York, 1999
- Balari, S. and L. Dini** (eds.) *Romance in HPSG*, CSLI, Stanford, 1997
- Barbu, A.-M** *Complex verbal*, manuscript, 1998, (to appear in *Studii și Cercetări Lingvistice*)
- Barwise, J. and J. Perry**, *Situations and Attitudes*, Bradford Books, MIT Press, Cambridge, MA, 1983
- Black, J. R. and V. Montapanyane** (eds.) *Pronouns, Clitics and Movement*, John Benjamin, Publishing Company, Amsterdam / Philadelphia 1998
- Borer, H.** *Parametric Syntax*, Foris, Dordrecht, 1984
- Borer, H.** (ed.), *Syntax and Semantics*, vol.19, *The Syntax of Pronominal Clitics*, Academic Press, Inc., Harcourt Brace Jovanovich, Publishers, Orlando, 1986
- Bouma, G. R. Malouf and I. A. Sag**, *Satisfying Constraints on Extraction and Adjunction*, *Natural Language and Linguistic Theory*, 19, 2001, 1-65
- Carpenter, B.** *The Logic of Typed Feature Structures*, Cambridge University Press, New York, 1992
- Cornilescu, A.** *A Note on Dative Clitics and Dative Case in Romanian*, *Revue Roumaine de Linguistique*, tome XXXII, No. 3, 1987, p. 213-225
- Dobrovie-Sorin, C.** *The Syntax of Romanian*, Mouton De Gruyter, Berlin, New York, 1994
- Farkas, D.** *Direct and Indirect Object Reduplication in Romanian*, *Proceedings of the 14th Regional Meeting of the CLS*, Chicago, 1978, p. 88-97
- Gerlach, B. and J. Grijzenhout** (eds.) *Clitics in Phonology, Morphology and Syntax*, John Benjamins Publishing Company, Amsterdam / Philadelphia, 2000
- Gierling, D.** *Clitic Doubling, Specificity and Focus in Romanian*, in Black and Montapanyane (1998), p.63-85
- Ginzburg, J. and I. A. Sag**, *Interrogative Investigations*, CSLI, Stanford, California, 2000.
- Green, G.** *The Structure of CONTEXT*, ms, 1997
- Green, G.** *Modeling Grammar Growth: Universal Grammar without Innate Principles or Parameters*, ms, 2000
- Gutiérrez-Rexach, J.** *The Formal Semantics of Clitic Doubling*, manuscript, 1999
- Ionescu, E.** *Accusative Weak Pronouns in Romanian*, in *Cahiers de Linguistique Théorique et Appliquée*, tomes XXXII-XXXIII, 1995-1996, p. 40-52
- Jaeggli, O.** *Three Issues in the Theory of Clitics: Case, Doubled NPs, and Extraction*, in Borer (1986), p.15-42
- Kathol, A.** *Linear Syntax*, OUP, Oxford, 2000
- King, J. P.** *A Logical Formalism for HPSG*, Ph.D. thesis, ms, 1989

- Müller, Ph. and I. A. Sag** *French Clitic Movement without Clitic or Movement*, *Natural Language and Linguistic Theory*, 15 (3), 1997, p. 573-639
- Monachesi, P.** *Decomposing Italian Clitics*, in Balari and Dini (1997) p. 301-354
- Monachesi, P.** *Clitic Placement in the Romanian Verbal Complex*, in Gerlach and Grijzenhout (2000), p. 255-294.
- Pollard, C and I. A. Sag**, *Information-based Syntax and Semantics*, CSLI, Stanford, California, 1987
- Pollard, C. and I. A. Sag** *Head-driven Phrase Structure Grammar*, The University of Chicago Press, Chicago, 1994
- Pollard, C and Eung Jung Yoo**, *A Unified Theory of Scope for Quantifiers and WH-Phrases*, *Journal of Linguistics*, 34, 1998, 415-445
- Reape, M.** *Domain Union and Word Order Variation in German*, in J. Nerbonne, K. Netter and C. Pollard (eds.), *German in Head-driven Phrase Structure Grammar*, CSLI, Stanford California, 1994, 151-197
- Sag, I. A. and Ph. Müller**, *French Clitic Movement without Clitic or Movement*, *Natural Language and Linguistic Theory*, 15 (3), 1997, 573-639
- Sag, I. A. and Th. Wasow**, *Syntactic Theory: A Formal Introduction*, CSLI, Stanford California, 1999
- Sanfilippo, A.** *Thematically Bound Adjuncts*, in Balari and Dini (1997) p. 355-391
- Shieber, St.** *An Introduction to Unification-based Approaches to Grammar*, CSLI, Stanford California, 1986
- Vallduvi, E. and E. Engdahl**, *Information Packaging and Grammar Architecture*, in J. N. Beckman (ed.) *Proceedings of the North East Linguistic Society*, University of Pennsylvania 1995, 519-533.

The RORIC-LING Bulletin

months 1 - 6

A number of **141** questions have been asked, by subscribers coming mostly from Romanian academic units, but not only. Software companies, both from Romania and from abroad, are also represented. Some of these questions have been asked more than once, as will be specified in the bulletin. There are four main categories of questions, corresponding to the topics of the BALRIC-LING Romanian part of the project, as follows:

- general questions;
- questions referring to Dependency Grammars and to the DGA annotation tool;
- questions referring to the dependency relations which have been established as being typical for the Romanian language, as well as referring to the annotation process of Romanian texts by means of DGA;
- questions referring to the HPSG formalism.

The questions have been grouped according to the topic they refer to (and not according to the subscribing date, namely not in chronological order). All information concerning the names and personal data of subscribers is stored in the RORIC-LING files but has been deleted from the bulletin, in order to facilitate reading and using this material, as well as the search process according to topic.

A number of **53** questions and corresponding answers, which refer strictly to the Romanian language and to the dependency relations which were established as being typical for Romanian, have not been translated into English and are not included in this version of the bulletin. They can be seen, together with the corresponding answers, in the Romanian version of the bulletin.

Some statistics:

Questions: 141

Country:	Romania	Other*
Questions:	107	34

Native Language:	Romanian	Other
Questions:	121	20

Activity Field:	Education	Research	Software Industry	Other
Questions:	89	18	12	22

* AUSTRALIA, AUSTRIA, CANADA, FRANCE, GERMANY, GREECE, ITALY, NETHERLANDS, TURKEY, UNITED KINGDOM, UNITED STATES

General Questions

What is the exact aim of this work?

From the way you are asking your question it is not clear to us whether you refer to a specific paper, program etc. that our Regional Information Center has displayed on the web or to the objectives of the entire project. That is why we are taking the liberty to answer you from a very broad perspective. If you will feel the need for certain details or for further clarifications, please don't hesitate to contact us once again.

The main objective of the BALRIC-LING project is to raise the awareness concerning the potential of the most advanced Human Language Technologies (HLT) and the possible scientific and industrial applications of the corresponding linguistic resources. Raising this awareness is necessary especially in the Balkans, where the fields of Natural Language Processing and Computational Linguistics are less known. Since HLT is a rather broad field, BALRIC-LING will focus on four topics only: word-centered linguistic resources, corpora and tagging, relevant supporting tools and possible advanced HLT usages of the first two.

In order to raise the awareness concerning these topics especially in Bulgaria and Romania, within the BALRIC-LING project two Regional Information Centers have been created in these countries. The Romanian Regional Information Center is called RORIC-LING and it will concentrate on the topics mentioned in the center's web page, that corresponding to the web address you have subscribed from.

An issue of great interest now-days is that of creating linguistic resources in general, corpora in particular, since, without these, advanced HLT applications can not exist. Within the first discussed topic, RORIC offers an annotation tool for corpus creation, which can be used within the formal framework of Dependency Grammars. Its value resides mainly in the fact that it is language independent. Many examples (annotated texts) of using this tool in the case of the Romanian language are offered, since this is of special interest to users in our country.

The other project partners will refer to different topics, all within the general themes previously mentioned. In order to find out more details about the BALRIC-LING project and to visit the web pages of the other project partners, you can visit the project home page at <http://www.lml.bas.bg> from where you must choose BALRIC-LING. Thank you for subscribing and for your interest in our project.

Hallo RORIC-LING team! An interesting project! I am a computer scientist and my question is: to what extent do I need other specific knowledge (for instance the grammars corresponding to various foreign languages) in order to deeply understand the concepts and issues discussed within this project?

In our view it is only necessary to understand the concepts and the theory that we have already presented on the web. If you will ever need or wish to apply these linguistic theories in connection with a given language, in the framework of various NLP applications (for instance parsing), like we have already done for Romanian, you will need to ask for linguistic guidance. Linguists will provide the necessary information concerning the application of the general theory with respect to a specific language. In order to find out more concerning the various possible computer science type applications of these theories, we recommend that you look over the virtual bulletin that will be published by us on the web, at the end of February.

Congratulation for the daring project you've got involved in. Hope you get along successfully. Does this project involve a text processing procedure and if it does what is the approach you chose: the more classical one based on the proposition calculus (Chomsky) or you treat it as a text classification problem? If the latter alternative is utilized what are the feature extraction and learning strategies you intend to employ?

Thank you for subscribing and for your interest in our project. The project will not involve a text processing procedure, at least not at this early stage. BALRIC-LING is mainly an awareness project having as main objective to raise the awareness concerning HLT mainly in the Balkan area. The first part of the project focuses on word-centered linguistic resources, corpora and tagging, relevant supporting tools. In case the project will be prolonged future topics might include a text processing procedure, which most likely will not involve a classical Chomskian approach.

What is the connection between the materials which are now on the web and the two topics which will follow?

The connection between this part of the project and the last one will become obvious when we will focus on establishing a theoretical specification for a morphological model for Romanian. The second topic proposed by RORIC and referring to WordNet represents a completely different topic. Actually, the main goal of the virtual seminar organized by RORIC is that of taking into account essential topics which are not linked to one another but which all refer to some

main aspects of language technology: word-centered linguistic resources and annotation; corpora and tagging; relevant supporting tools.

How long will this project go on? Is the HLT project only especially for IT companies and IT people or for everyone? I'm interested in more information about the HLT project.

The BALRIC-LING project has started on Sept.1, 2001 and will go on for 18 months (unless it will be prolonged). It is funded by the European Commission.

The main objective of the BALRIC-LING project is to raise the awareness concerning the potential of the most advanced Human Language Technologies and the possible scientific and industrial applications of the corresponding linguistic resources. Raising this awareness is necessary especially in the Balkans, where the fields of Natural Language Processing and Computational Linguistics are less known. Since HLT is a rather broad field, BALRIC-LING will focus on four topics only: word-centered linguistic resources and annotations, corpora and tagging, relevant supporting tools and possible advanced HLT usages of the first two.

In order to raise the awareness concerning these topics especially in Bulgaria and Romania, within the BALRIC-LING project two Regional Information Centers have been created in these countries. The Romanian Regional Information Center is called RORIC-LING and it will concentrate on the topics mentioned in the center's web page, that corresponding to the web address you have subscribed from.

As of today you will find more details concerning the BALRIC-LING project in the RORIC-LING home page, as well.

The other project partners will refer to different topics, all within the general themes previously mentioned. Not all partners have loaded their web pages yet, but will do so soon. In order to find out more details about the BALRIC-LING project and to visit the web pages of the other project partners, you can visit the project home page at <http://www.lml.bas.bg> from where you must choose BALRIC-LING.

The project does not refer only to the Balkan area, it addresses people dealing with HLT from everywhere. Also, it is not only for IT companies and IT people. The project tries to raise the general awareness concerning this field and we will be happy to answer questions coming from all those who are - or become - interested. Thank you for subscribing and for your interest in our project.

Questions Referring to Dependency Grammars and DGA

I understand that DGA saves results in an XML format. Do you also have an XSLT which can transform results from the XML format into a different XML format and, if yes, which one? If not, does this mean that it is the user's job to write an appropriate XSLT?

The XML format used by DGA is a very simple one, inspired by the XCES standard. The needs of the user can, however, vary a lot. Therefore, if the user requires a different format, he must write a XSLT by means of which the corpus is transformed from the XML format used by DGA into the required format. For instance, I use a XSLT that turns the texts annotated with DGA into the HTML format which allows viewing these texts on the web.

What are the advantages of using DGA?

The main advantages of using DGA derive from the fact that it represents a tool which is language independent. Also, it was designed to be independent from the various formalisms related to Dependency Grammars. Other important advantages of DGA derive from its characteristics mentioned in the user manual: usage easiness, portability, conformity with up-to-date standards, flexibility.

What are the most important differences between Dependency Grammars and Phrase-structure Grammars?

As it is well known, there are two diametrically opposed methods of describing the syntactic structure of natural sentences: dependency (D-)trees and phrase-structure (PS-)trees. Obviously, combinations of the two methods are possible, with lines of compromise being drawn at different points; but there is no essentially distinct third possibility.

There are five major respects in which D-language is different from PS-language:

A first significant difference refers to constituency vs. relations. A PS-tree of a natural-language expression shows which items of the latter – word forms or phrases – "go together" (i.e., combine) with which other items to form tight units of a higher order. A PS-tree reveals the structure of an expression in terms of groupings of its actual elements: maximal blocks, which consist of smaller blocks,

which consist of still smaller blocks, etc. The PS- approach concentrates on constituency, the main logical operation in this approach being set inclusion (to "belong to a phrase" or "belong to a category"). Under the PS-approach, an actual sentence is, so to speak, cut into (generally two) major constituents, each of which is subsequently cut in its turn, etc. This approach thus favors the analytical viewpoint. A D-tree, on the other hand, shows which items are related to which other items and in what way. The D-approach concentrates on the relationships between ultimate syntactic units, i.e., word forms. The main logical operation here is the establishing of binary relations. Under the D-approach, an actual sentence is, so to speak, built out of words, linked by dependencies. This approach thus favors the synthetic viewpoint.

Another difference between PS-language and D-language refers to the fact that, in a PS-tree, the syntactic class membership (i.e., categorization) of an item is specified as an integral part of the syntactic representation. Symbols like NP, VP, N, PP, etc. appear in PS-trees as labels on nodes. In other words, distributional properties of syntactic units (i.e., the traditional parts of speech and syntactic features, rechristened 'categorization' and 'subcategorization') are used as the main tool to express their syntactic roles. In a D-tree, on the other hand, the symbols representing the syntactic class membership and other syntactic properties of an item are not admitted as immediate elements of syntactic structure. (Such information is included in the dictionary, lexicon etc.).

A third important difference refers to terminals and non-terminals. In a PS-tree, most nodes are non-terminals: they represent syntactic groupings or phrases and do not correspond to the actual word forms of the sentence under analysis. A D-tree, on the contrary, contains terminal nodes only; no abstract representation of groupings is needed.

The linear order of nodes generates another difference: In a PS-tree, nodes must be ordered linearly. The order is not necessarily that of the actual word forms of the sentence, but some linear order is unavoidable. The PS-language is essentially linear. In a D-tree, on the other hand, the nodes are in no linear order at all. The linear order of word forms in the sentence is an expressive means used by actual languages to encode something different from this order itself, namely syntactic relations, and therefore, linear order should not be present in syntactic structures. The D-language is essentially two-dimensional. Finally, a PS-tree does not specify the type of syntactic link existing between two items (and cannot do so, at least not in a natural and explicit way). A D-tree, on the other hand, puts particular emphasis on specifying in detail the type of any syntactic relation obtained between two related items.

How could a corpus obtained by annotation with DGA be used?

A possible utilization of such a corpus would be that of performing syntactic analysis (parsing). This has already been done at the University of Bucharest, for Romanian, within the DBR-MAT project, funded by the Volkswagen Foundation.

The solution which was successfully chosen in the case of the Romanian language within the framework of the DBR-MAT project, for performing "dependency parsing", is of stochastic nature. It consists of associating a probability to each syntactic structure, and of choosing that syntactic structure which has the maximum associated probability corresponding to each given sentence. Assigning such a probability means finding a stochastic model, namely the most adequate one. According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web. The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies). We will get back to you on how these sets were found, if you are interested in the stochastic aspects of this approach.

Do you consider the Romanian language more suited to a scientific approach using Dependency Grammars rather than Phrase-Structure Grammars? (asked twice)

Yes, mainly because this approach is closer to the traditional way of performing syntactic analysis. This fact probably determines the Romanian linguists to feel much closer to this approach, which they have already successfully applied relatively to the Romanian language, by performing "dependency parsing", within the DBR-MAT project, which was funded by the Volkswagen Foundation (1996-1998).

Do corpora for the Romanian language, containing texts annotated according to the Dependency Grammar formalism, exist? (asked twice)

The creation of such a corpus has been initiated now, within the BALRIC-LING project framework. The annotated texts which already exist on the web are part of this corpus and their number will increase in the near future.

Is Link Grammar a dependency type grammar? (asked twice)

Link Grammar is of dependency type but much more lexicalized. A formal grammatical system called a "link grammar" requires that a sequence of words is in the language of a link grammar if there is a way to draw links between words in such a way that (1) the local requirements of each word are satisfied, (2) the links do not cross, and (3) the words form a connected graph. The formalism is lexical and makes no explicit use of constituents and categories.

Link Grammars resemble Dependency Grammars and Categorical Grammars. There are also significant differences, the most important aspect being the fact that Link Grammars are much more lexicalized.

What parsing algorithms using Dependency Grammars exist? Has your group already used any of them?

Syntactic parsing has been performed, within the formal framework of Dependency Grammars, using "Constraint Dependency Grammar" - CDG (Maruyama, 1990). Decision procedures for dependency parsing using graded constraints exist. CDG strictly separates possible structural descriptions from the correctness conditions for linguistic structures. CDG is weakly context-sensitive. In order to learn about CDG based algorithms, we recommend reading

Menzel, Wolfgang and Schroder, Ingo, "Decision procedures for dependency parsing using graded constraints", in: Sylvain Kahane si Alain Polguere, editors, "Proc. Coling - ACL Workshop on Processing of Dependency-based Grammars", pag. 78-87, Montreal, Canada, 1998.

As far as our group is concerned, we have performed dependency parsing using a stochastic approach. It consists of associating a probability to each syntactic structure, and of choosing that syntactic structure which has the maximum associated probability corresponding to each given sentence. Assigning such a probability means finding a stochastic model, namely the most adequate one.

According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web.

The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P, finding set D is no longer a problem.

Does a Dependency Grammar for the Romanian language exist? (asked twice)

Within the framework of this linguistic theory specifying a Dependency Grammar means finding a set of constraints which can lead to establishing that certain syntactic structures are correct, while others are not. For instance, according to such constraints, it can be decided that certain words of a sentence may be head-words, while others may not. In other words, specifying a Dependency Grammar for a specific language means finding a set of rules which indicate what dependency relations are accepted in that language. This has not been done yet, for Romanian, therefore a Dependency Grammar for this language does not exist. Only specific dependency relations have been established, dependency relations which can be used for performing various tasks, such as dependency parsing.

What are the possible types of dependencies?

The classical types of dependencies are those of type subject, object and complement. However, these dependencies can be further refined. For instance, in establishing the most frequent types of dependencies for Romanian we have, in most cases, considered the syntactic function (as in classical syntactic analysis) of

the dependent. A table with the most frequent dependency relation types occurring in Romanian can be found in the paper which is now on the web and will be updated by RORIC at the end of February.

Please give an example of how a corpus obtained by DGA annotation can be used

An example of how such a corpus can be used is that of performing stochastic parsing. Our group has already performed "dependency parsing" of Romanian sentences in a stochastic manner. According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web. The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P, finding set D is no longer a problem.

Do you recommend performing stochastic parsing using Dependency Grammars or Generative Grammars?

We recommend stochastic parsing based on Dependency Grammars since our group has already successfully performed it in the case of the Romanian language. According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web. The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P, finding set D is no longer a problem.

Which were discovered first, Dependency Grammars or Generative Grammars? Please present a short history of these two types of grammars.

The main stages in the evolution of these types of grammars are the following:

1. Panini (2600 years ago, India) recognized, distinguished and classified semantic, syntactic and morphological dependencies.
2. The Arabic grammarians (1200 years ago, Iraq) recognized government and syntactic dependency structures.
3. The Latin grammarians (800 years ago) recognized 'determination' and dependency structures.
4. School grammars of English in Europe and U.S.A. taught sentence-analysis in terms of dependency, and the 'sentence diagramming' which has been popular since the late 19-th century (using a system invented in U.S.A.) is DG.
5. Lucien Tesniere (1930s France) developed a relatively formal and sophisticated theory of DG grammar for use in schools. This bottom-up approach is still widely used in Europe, and by Russians and slavists in U.S.A.
6. In 1933 Leonard Bloomfield in the U.S.A. developed a top-down approach: Immediate Constituent Analysis, which turned into PSG ('phrase-structure grammar').

The popularity of dependencies as a formal way of representing the syntactic structure of sentences has been constantly growing and has culminated with the work of Lucien Tesniere from 1959. In spite of this, however, at the beginning of the 30s, in North America, the 'immediate constituency' syntax started replacing the 'dependency syntax'. The first later turned into 'PS analysis' which determines the

phrase-structure of a sentence. Rigorously stated by Leonard Bloomfield (Bloomfield 1933), but also by Wells in 1974 and Percival in 1976, the PS-type representation in syntax has been promoted with great energy by the structuralist school in the 30s, 40s and 50s. It became the only syntactic representation seriously taken into consideration by Noam Chomsky and the generative school which he founded at the end of the 50s.

Can you give an example of another type of grammars (besides Dependency Grammars) which provide an appropriate description of natural languages?

Another class of grammars which provide an appropriate description of natural languages is that of Contextual Grammars, which also do not fit into the Chomsky hierarchy. Contextual Grammars were introduced by Solomon Marcus in 1969 as "intrinsic grammars" without auxiliary symbols, based only on the fundamental linguistic operation of inserting words into given phrases according to certain contextual dependencies. Contextual Grammars include contexts, i.e. pairs of words, associated with selectors (sets of words). A context can be adjoined to any associated selector element. In this way, starting from a finite set of words (axioms), the language is generated. It has been shown that this formalism provides an appropriate description of natural languages. It is only in 1999 that K. Harbusch presents a parser based on Contextual Grammars. Recent very encouraging results have determined researchers to concentrate on defining a Contextual Grammar for English. For more information on Contextual Grammars see:

1. **S.Marcus, C.Martin-Vide, G.Paun.** Contextual Grammars as Generative Models of Natural Languages. Computational Linguistics, 24(2), p. 245-274.
2. **F.Hristea.** Introducere in procesarea limbajului natural cu aplicatii in Prolog. Editura Universitatii din Bucuresti, 2000, p. 102-113 (in Romanian).

Are Dependency Grammars and Contextual Grammars one and the same type of grammars?

NO. Contextual Grammars represent a different class of grammars which provide an appropriate description of natural languages and which also do not fit into the Chomsky hierarchy. Contextual Grammars were introduced by Solomon Marcus in 1969 as "intrinsic grammars" without auxiliary symbols, based only on the fundamental linguistic operation of inserting words into given phrases according to certain contextual dependencies. Contextual Grammars include contexts (pairs of words), associated with selectors (sets of words). A context can be adjoined to any associated selector element. In this way, starting from a finite set of words (axioms), the language is generated. It has been shown that this formalism provides an appropriate description of natural languages. It is only in 1999 that K. Harbusch

presents a parser based on Contextual Grammars. Recent very encouraging results have determined researchers to concentrate on defining a Contextual Grammar for English. For more information on Contextual Grammars see:

1. **S.Marcus, C.Martin-Vide, G.Paun.** Contextual Grammars as Generative Models of Natural Languages. *Computational Linguistics*, 24(2), p. 245-274.
2. **F.Hristea.** Introducere in procesarea limbajului natural cu aplicatii in Prolog. Editura Universitatii din Bucuresti, 2000, p. 102-113 (in Romanian).

Has your group ever performed syntactic parsing using Dependency Grammars and how?

Our group has performed "dependency parsing" using a stochastic approach. According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web.

The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P, finding set D is no longer a problem.

Which of the two classes of grammars (Dependency and Generative grammars respectively) express best natural language phenomena?

The answer to this question depends on what we mean by Generative Grammars. This is an extremely broad class of grammars within which various formalisms for describing natural language phenomena exist. Dependency Grammars have also been formalized in various ways.

That is why we shall try to answer the above question by viewing the issue from three different points of view, namely:

1. From a formal point of view. This point of view refers to the generative capacity of a specific class of grammars. In order to be considered adequate, a class of grammars must be sufficiently restrictive so that it does not allow the generation (description) of any type of language, but it must also be sufficiently powerful in order to allow the description of natural language phenomena. From this point of view we consider that the two mentioned classes of grammars are equivalent.

After having accepted the fact that natural language phenomena are too complex for the descriptive capacity of context independent grammars, the class of "mildly context-sensitive languages" has recently been taken into consideration. This class of languages is usually accepted as being sufficiently adequate for describing natural language and is generated by a variety of grammatical formalisms (considered independently and for various reasons):

K. Vijay-Shanker, D.J. Weir, The Equivalence of Four Extensions of Context-Free Grammar. *Math. Systems Theory*, 27, 1994.

As far as Dependency Grammars are concerned, formalisms which make them equivalent to context independent grammars exist

H. Gaifman, Dependency systems and phrase-structure systems. *Information & Control*, 8, 1965.

but other formalisms, which allow them to describe mildly context-sensitive languages, exist as well:

H. Maruyama, Constraint dependency grammar and its weak generative capacity. *Computer Software*, 1990.

2. From a linguistic point of view. This point of view refers to the easiness with which a linguist can describe linguistic phenomena typical of a specific language using a specific formalism. From this point of view we think the answer to your question depends on the considered language and on the linguistic tradition which exists for that specific language. With respect to the Romanian language we consider the Dependency Grammar formalism more adequate since it is closer to

the traditional way of performing syntactic analysis of Romanian and therefore permits more easily to incorporate knowledge provided by Romanian linguists.

3. From the point of view of natural language stochastic modeling. For a complete discussion concerning the advantages provided by Dependency Grammars in stochastic modeling of natural language see section 12.1.7 of

C. D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing. The MIT Press, 1999.

For the moment we shall only mention the fact that the best stochastic parsing system known until now is based on Dependency Grammars:

M. J. Collins, Three generative, lexicalised models for statistical parsing. ACL 35, 1997.

Does your group have any contributions to the theory of Dependency Grammars or regarding the way of using them?

Our group has performed stochastic parsing using Dependency Grammars. According to our stochastic approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web. The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P, finding set D is no longer a problem.

Have you heard of a parser based on Dependency Grammars? Have you ever performed parsing using Dependency Grammars?

Our group has already performed "dependency parsing" of Romanian sentences in a stochastic manner. According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web. The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P , finding set D is no longer a problem.

What is detecting dependencies relevant for?

Detecting dependencies is relevant mainly because, at the heart of sentence structure are the relations among words, no matter if by these relations we mean the possible grammatical functions or the links which bind words into larger units (like phrases, for instance). Unlike generative grammars, dependency grammars can describe linguistic phenomena like the variation of word order within a sentence or the existence of discontinuous constituents more successfully. As far as applications are concerned, the formalism of dependency relations has been proven to be more adequate than the generative grammars formalism for being combined with maximum entropy modeling in order to obtain a stochastic parser, for instance.

What other types of grammars are used in NLP?

The classical top-down and bottom-up parsing algorithms are based on generative grammars, which view the sentence structure as being formed of constituents. In this case, the structure of a sentence, given by its constituents, represents the main concept of syntax. Unlike generative grammars, dependency grammars are not based on the notion of constituent but on the direct relations existing among words. The dependency structure can be viewed, among other ways, versus constituent structure. The main idea behind the notion of dependency is that each word depends on the word which links it to the rest of the sentence, practically explaining why it is used. Unlike generative grammars, dependency grammars can describe linguistic phenomena like the variation of word order within a sentence or the existence of discontinuous constituents more successfully.

Another class of grammars which generates languages not having a direct link to the Chomsky hierarchy is that of Contextual Grammars. Contextual Grammars were introduced by Solomon Marcus in 1969 as "intrinsic grammars" without auxiliary symbols, based only on the fundamental linguistic operation of inserting words into given phrases according to certain contextual dependencies. Contextual Grammars include contexts i.e. pairs of words, associated with selectors (sets of words). A context can be adjoined to any associated selector element. In this way, starting from a finite set of words (axioms), the language is generated. It has been shown that this formalism provides an appropriate description of natural languages. It is only in 1999 that K. Harbusch succeeds in presenting a parser based on contextual grammars. Recent promising results have encouraged researchers to concentrate on building a contextual grammar for English.

Other types of grammars used in NLP are Link Grammars and Tree Adjoining Grammars, as well as others. Please feel free to contact us again if you are interested in finding out more about a specific class of grammars.

How did you use the set of tags XCES when designing the DGA tool? What does the resemblance consist in?

Since a standard set of tags for syntactic annotation of a text does not exist yet, DGA uses a set of tags inspired by XCES (the standard set of tags for representing morphosyntactic annotation) in order to represent annotated texts. The general idea was that of using a set of tags as simple as possible so that it can easily become compatible with a future standard. From XCES we have kept those tags indicating the general structure (sentence delimitation by <s>...</s>, delimitation of each token within a sentence by <tok>...</tok>). Corresponding to each token we have

kept the marking of the orthographic form by <orth>...</orth> and of the non-ambiguous part of speech by <ctag>...</ctag> . We have given up those XCES tags which refer to morphological information and we have created new tags corresponding to syntactic information: <syn>...</syn>, <head>...</head>, <reltype>...</reltype>.

How can one view on-line the XML files resulting after DGA annotation? (asked twice)

There are various possible solutions to this problem. In what follows, we are presenting a solution that has already been implemented:

First of all, the XML files resulting as a consequence of DGA annotation have been transformed, using XSLT, into HTML files. Within the HTML files each sentence is contained in a FORM. As a consequence of the SUBMIT operation (in our case click on a sentence), the FORM will send the server (by means of some fields of type HIDDEN) the information contained in the annotation. Based on this information a perl script existing on the server builds a jpeg image that represents the annotation in the usual graphical form. This image is returned to the browser, which will display it in a new window. You can see how this works at the address:

<http://phobos.cs.unibuc.ro/oric/texts/indexro.html>

What would defining a Dependency Grammar for a specific language require? Does such a grammar exist for Romanian?

Defining a Dependency Grammar for a specific language requires finding a set of constraints that can help in establishing the fact that certain syntactic structures are correct, while others are not. For instance, according to such constraints one can decide that certain words of a sentence can be head words, while others can not. In other words, specifying a Dependency Grammar for a specific language means finding a set of rules which indicate what dependency relations are accepted in that language. Defining such constraints and rules corresponding to a specific language can be a very difficult and time consuming task. This is probably one of the reasons why a Dependency Grammar for Romanian does not exist yet. Our group has established a set of dependency relations in Romanian, relations that can be used in performing various tasks, such as stochastic parsing of Romanian sentences.

What parsing algorithms for Dependency Grammars exist?

Syntactic parsing has been performed, within the formal framework of Dependency Grammars, using "Constraint Dependency Grammar" - CDG (Maruyama, 1990). Decision procedures for dependency parsing using graded constraints exist. CDG

strictly separates possible structural descriptions from the correctness conditions for linguistic structures. CDG is weakly context-sensitive. In order to learn about CDG based algorithms, we recommend

Menzel, Wolfgang and Schroder, Ingo, "Decision procedures for dependency parsing using graded constraints", in: Sylvain Kahane si Alain Polguere, editors, "Proc. Coling - ACL Workshop on Processing of Dependency-based Grammars", pag. 78-87, Montreal, Canada, 1998.

Our group has also performed dependency parsing, using a stochastic approach. This approach consists of associating a probability to each syntactic structure, and of choosing that syntactic structure which has the maximum associated probability corresponding to each given sentence. Assigning such a probability means finding a stochastic model, namely the most adequate one. According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web.

The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P, finding set D is no longer a problem.

Give an example of how the Dependency Grammar formalism can or has been used in the case of the Romanian language.

The formalism of Dependency Grammars has been used, within the DBR-MAT project, in the case of the Romanian language, in order to perform stochastic parsing of Romanian sentences. Within the formal framework of Dependency Grammars finding a parsing algorithm means finding an algorithm which has as

input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as in the paper which now exists on the web. The steps of such an algorithm are: finding set T ("part of speech tagging"), finding set P (namely the dependency relations) and finding set D (namely the types of the dependencies). The main conclusion that we came to, within the DBR-MAT project, and independently of any specific language, was that the formalism of Dependency Grammars is extremely adequate and can be successfully used in performing stochastic parsing.

What are the main theories in the DG family?

The main theories in the DG family are the following ones:

- Case Grammar (Anderson)
- Daughter-Dependency Theory (Hudson)
- Dependency Unification Grammar (Hellwig)
- Functional-Generative Description (Sgall)
- Lexicase (Starosta)
- Meaning-Text Model (Melcuk)
- Metataxis (Schubert)
- Unification Dependency Grammar (Maxwell)
- Constraint Dependency Grammar (Maruyama)

How is DG different from PSG?

As Richard Hudson points out, grammars, and theories of grammar, can be classified according to whether the basic unit of sentence structure is:

- the phrase (PSG);
- the dependency between two words (DG).

Each approach implies the other:

- PSG implies inter-word dependencies (but only if one word is chosen as the phrasal head);
- DG implies phrases (a word plus its dependents and their phrases constitutes a phrase).

Is DG just a notational variant of PSG?

A number of logicians, including Bar-Hillel, proved that DG (including Categorical Grammar) is WEAKLY equivalent to a context-free PSG (Gaifman, "Dependency

systems and phrase-structure systems"). This result is generally accepted. But it is NOT a notational variant of PSG because it is not STRONGLY equivalent - i.e., as Richard Hudson points out, it does not allow the same analyses:

- Phrases are implicit, not explicit so
 - phrases can not be classified separately from their heads;
- relations are explicit, not implicit, so
 - relations can be classified and labeled;
- all phrases must be endocentric, so
 - apparently exocentric constructions such as gerunds (an NP with the internal structure of a clause) are a fundamental challenge;
- no non-terminal nodes are allowed, so DG does not allow
 - unary branching (e.g. NP consisting of just N),
- VP (contrasting with V).

What is Word Grammar? Is it related to Dependency Grammar?

Word Grammar is a theory that Richard Hudson has been developing since early 1980s. It is firmly based on DG and probably has the best possible combination of other features. The main features of WG as noted by its author are:

- it is monostratal - there is only one syntactic structure per sentence, paired with a semantic structure and a phonological one;
- it is enriched - it allows multiple dependencies (a word can have more than one parent);
- it generalizes by means of default inheritance based on the 'isa' relationship;
- it allows labeled relationships (in an isa hierarchy);
- it is non-modular and cognitive: language is an area of the general network of knowledge.

Does the DGA system rely completely on the user in the annotation process, or is it a semi-automatic system that gives the user options to choose from? If it is not semi-automatic (as it seems to me), why is it not so? Even if the system starts with no knowledge whatsoever, in time it could at least accumulate patterns to reduce the onus on the user.

DGA is not semi-automatic in the sense that it does not contain an internal mechanism for initially annotating a corpus in order to present it to the user for performing corrections. DGA has been thus designed in order to make it language independent and as independent as possible from all types of formalisms related to Dependency Grammars. However, DGA can easily be turned into a semi-automatic tool by integrating various external products (POS tagger, parser etc.). DGA allows viewing and modifying previously annotated corpora (by using the Open corpus command of menu File). Although, initially, this facility was intended to be used in order to modify annotations performed by means of DGA, it can also be used in the case of various external products (POS tagger, parser etc.). In order to do this it is only necessary for the automatically annotated corpus (with the external product) to be turned from the format used by the corresponding external product into the XML format used by DGA. After performing this operation the corpus can be opened with DGA and the automatically performed annotations can be corrected.

What are possible software applications of the presented topics (e.g. dependency grammar)? Are there existing programs for the English language that could be ported to Romanian once corresponding rules/descriptions/annotations/whatever have been developed for the Romanian language?

One example of Dependency Grammar - based applications is parsing, the software applications being the corresponding parsing programs.

Syntactic parsing has been performed, within the formal framework of Dependency Grammars, using "Constraint Dependency Grammar" - CDG (Maruyama, 1990). Decision procedures for dependency parsing using graded constraints exist. CDG strictly separates possible structural descriptions from the correctness conditions for linguistic structures. CDG is weakly context-sensitive. In order to learn about CDG based algorithms, we recommend reading

Menzel, Wolfgang and Schroder, Ingo, "Decision procedures for dependency parsing using graded constraints", in: Sylvain Kahane si Alain Polguere, editors, "Proc. Coling - ACL Workshop on Processing of Dependency-based Grammars", pag. 78-87, Montreal, Canada, 1998.

As far as our group is concerned, we have performed dependency parsing using a stochastic approach. It consists of associating a probability to each syntactic structure, and of choosing that syntactic structure which has the maximum associated probability corresponding to each given sentence. Assigning such a probability means finding a stochastic model, namely the most adequate one. According to this approach, in order to find the syntactic dependency structure of a given sentence it is not necessary to explicitly specify a Dependency Grammar. The grammar will be implicitly included in the parameters of the stochastic model, which, in turn, will be estimated by means of the linguistic data (namely of a corpus).

Within this framework one can say that finding a parsing algorithm means finding an algorithm having as input a sentence and as output the syntactic structure (S,D) of that sentence, where $S=(T,P)$ and D have the same significance as that explained in the paper which is on the web.

The stages of such an algorithm are: finding set T ("part of speech tagging"); finding set P (namely finding the dependency relations); finding set D (namely establishing the types of the dependencies).

Set T was found using an algorithm proposed by Ratnaparkhi in 1996. This algorithm is of stochastic nature and uses maximum entropy. Set P was also found by means of a stochastic algorithm, namely the algorithm proposed by Eisner, also in 1996. We have modified this algorithm by changing the stochastic model and by again using maximum entropy. The algorithm for finding set P represents an implementation of the Dynamic Programming Method with the aim of finding the most probable parse in a bottom-up manner. After determining sets T and P, finding set D is no longer a problem.

The existing programs are language independent and have been successfully tested by us in the case of the Romanian language.

What is XCES? (asked twice)

XCES (XML Corpus Encoding Standard) is a standard for corpus representation. More information and details concerning XCES can be found at

<http://www.cs.vassar.edu/XCES>

Can DGA be modified, in principle, in order to be used for morpho-syntactic annotation as well? What would this operation require? (asked twice)

Yes, it can. In order to be used for morpho-syntactic annotation as well it is only necessary for DGA to allow the specifying of morphological information relatively to each word. This can be easily done by adding, in the contextual menu which is opened when performing a right click on a word, a command of type "morphology", for instance. When using this command you will be able to enter the morphological information corresponding to a specific word inside a dialog box which will open up.

Does the possibility of assisting the DGA annotation process by means of external products (POS tagger, parser etc.) exist? In this case the role of the user would be that of correcting an automatically performed annotation. Therefore the speed of the annotation process would increase.

Yes, this possibility exists. DGA allows viewing and modifying previously annotated corpora (by using the Open corpus command of menu File). Although, initially, this facility was intended to be used in order to modify annotations performed by means of DGA, it can also be used in the case of various external products (POS tagger, parser etc.). In order to do this it is only necessary for the automatically annotated corpus (with the external product) to be turned from the format used by the corresponding external product into the XML format used by DGA. After performing this operation the corpus can be opened with DGA and the automatically performed annotations can be corrected (modified).

Questions Referring to HPSG

Are there any computational implementations of HPSG?

Yes, there are. Among the most recent ones (known by us) is an implementation of the HPSG grammar of interrogatives in English. There is also a computational implementation of Andreas Kathol's analysis of word order phenomena in German.

Are any HPSG introductory courses being taught at the University of Bucharest?

Up to the past year, there were two courses of HPSG, both at the Faculty of Letters of the University of Bucharest: an introductory one (for the fourth year of studies), and another one with more applications to Romanian, for the master program in theoretical linguistics. By a decision of the head of the department of Romanian, the former course was eliminated, so during this academic year only one HPSG course is being delivered.

Is HPSG a Universal Grammar?

The ultimate roots of HPSG lie in Chomsky's program of the Universal Grammar. In this sense, HPSG is a version of the Universal Grammar, because it is naturally interested in the invariants of human language. Unlike Chomsky's program, though, HPSG does not privilege those invariants. On the contrary, HPSG also approaches those facts tight to the idiosyncratic aspects of languages. From this point of view, HPSG is rather close to 'traditional grammar'.

Do you plan to apply HPSG to Romanian?

We are already applying it. You will see this, in just a few months, right here, in this page. In fact, we are a small group of researchers analyzing Romanian from the HPSG perspective.

Analyze the sentence "The girl is laughing happy" in HPSG.

This is a phrase of type head-subject. The subject is the noun phrase "the girl", while the head is the verb phrase "is laughing happy". This latter phrase is of type head-adjunct. The head is the verb "is laughing" and the adjunct is the adjective "happy". English does not force the adjective to show agreement features between the adjective and the subject noun phrase "the girl". But, in a language like Romanian, this agreement is obligatory. HPSG resolves this further dependency in a very simple way: the adjective displays the information that its subject is identical to the subject of the phrase "is laughing happy".

What does "strict lexicalism" mean and is it typical for the HPSG approach to grammatical theory?

Strict (or strong) lexicalism is the view that the internal structure of the word is independent of the way the words contribute to the structure of the phrases. Strict lexicalism characterizes HPSG as well as other grammatical theories - for instance, the Minimalist Program also adheres to it, in spite of the fact that the previous version of Chomsky's program - The GB Theory - is not lexicalist.

How are long-distance dependencies treated in HPSG?

There are three components of the HPSG treatment of LDDs: a constraint on a particular type of phrase (i. e. the head-filler phrase), a constraint on phrases instantiating the feature SLASH, and a constraint ruling the collection of this feature on a lexical head (for details see the presentation on our web site, section 2.2.2.2). Thanks to these constraints, a 'gap' is 'collected' by a lexical head and it is allowed to percolate up to the point where it is 'filled' with the corresponding grammatical information:

(1) Bagels _i , John always said that he likes _i .

In (i) and (iv), likes and said, respectively, 'collect' the gap, in (ii)-(iii), and (v)-(vi) the gap successively 'percolates' different phrases, while (1) is the closure itself of the gap (through the constraint on the head-filler phrase):

- (i) likes _i
- (ii) he likes _i
- (iii) that he likes _i
- (iv) said that he likes _i .
- (v) always said that he likes _i
- (vi) John always said that he likes _i .

How can agreement phenomena be classified according to the HPSG theory? (asked twice)

Pollard and Sag ("Head-driven Phrase Structure Grammar", Chicago University Press, 1994, 73-88) classify agreement according to the elements involved in this relationship:

1. Pronoun-antecedent

2. Subject-verb
3. Determiner-noun

One must emphasize the fact that this classification does not follow from theory-internal reasons. That is, it is not determined by the fact that someone works in HPSG. This classification is determined by the nature of the language subject to investigation - in the present case, English.

Let us assume that we are analyzing an unknown language by means of a bilingual dictionary (which assigns a translation to each phrase of this unknown language). Is there any way to discover, thanks to this translation procedure, the relevant grammatical categories of the unknown language? To be more specific: assume that the language in question is English where, as known, the gender agreement between noun and adjective is unspecified. Assume also that English words are translated into Romanian by means of a dictionary. Can we imagine a procedure for writing the features of, say, adjectives in English starting from the features of adjectives in Romanian, along with the dictionary and the general principles of HPSG?

No, I don't think this is possible. If, for instance, we start up with the information about noun-adjective agreement in Romanian, without knowing anything about what happens in English concerning the same topic, we will probably be inclined to transfer the peculiarities of agreement in Romanian to English (which is, of course, a mistake).

May one assimilate unification to the union of set theory?

Yes, one may do so. What is unified can equally be regarded as a union.

Are there any other HPSG presentations in Romanian?

Yes, there are, for instance, in Doina Tatar, "Inteligența artificială", Editura Albastra, Cluj, 2001.

How could I find more works on HPSG? (asked twice)

Type hpsg as a key word and you will find an address for the site "HPSG literature", which contains bibliography. Many works are accessible on line.

Have any HPSG-based computational devices been implemented already?

I am not sure I understand correctly what you mean by "HPSG-based computational devices". If you refer to HPSG principles (for instance the (syntactic) principle of the HEAD features, which - in my terminology - is a HPSG-based computational device), then I would say that I know of no specific implementation. Nevertheless, he who wants to implement a certain fragment of a grammar, cannot avoid the implementation of the most general HPSG devices - or principles - like HFP. So, even if I am not able to indicate a specific work dealing with this topic, such an implementation must exist, and, for sure, it is feasible.

What programming language is used for computational applications of HPSG?

We know of PROLOG HPSG applications but we think that such applications in LISP also exist.

Does HPSG borrow anything from the modular architecture of computers? (asked twice)

Yes, it does. The way constraints are checked in HPSG resembles much the way you may perform independent operations when you use your computer. For instance, you are not bound to quit the program Word if you want to listen to a CD: you simply use the CD option and you continue working. Similarly, constraints are checked independently, that is, no order is a priori imposed.

What programming language do you recommend for computational applications of HPSG?

Most HPSG applications are programmed (especially in Europe) in PROLOG. This, of course, does not mean that one should underestimate LISP.

What linguistic phenomena are best modeled through HPSG?

HPSG considers itself to be a grammar of a language in general, so I would not say that there are "privileged" and "disgraced" phenomena (so to speak). Nevertheless, I might "reverse" the question and focus on what was felt as a failure: the analysis of unbounded dependencies (like in the sentence "Who do you think killed Kennedy ?") which was subject to a careful and long elaboration - the first version

not being satisfactory. Also, the analysis of relative clauses (which is a chapter of Unbounded Dependencies) enjoyed successive reconsiderations.

Default constraints and lexical rules are not declarative devices, but procedural ones. Under these conditions is it right to say that HPSG is not a purely declarative theory? (asked twice)

Yes, it is. Nevertheless, it has to be said that in the absence of these devices the analysis of certain phenomena in natural language would meet impressive difficulties.

What justifies the device of structure sharing? (asked twice)

Structure sharing expresses identities of information that are essential to the correctness of the construction. For instance, in the sentence "John, nobody believes that police looks for it" it is essential to point out that the NP John is at the same time the understood object of the verb to look for. On the contrary, the fact that nobody and police have the same person and number is not essential for the grammaticality of the sentence. Consequently, there is a distinction between the two identities and it is the duty of the theory to emphasize it. Structure sharing makes the difference.

How can one analyze in HPSG the sentence "The moon shines happy" ?

This structure is first a phrase of type head-subject (the head being the phrase "shines happy" and the subject being the NP "the moon ". The VP "shines happy", in turn, is a phrase of type head-adjunct. The verb is the head while the adjective is the adjunct. In Romanian, unlike English, this sentence shows the double dependency of the adjective - because of the visible agreement features of the adjective: the gender and the number of the adjective must be the same with the gender and the number of the NP "the moon". This double dependency is pointed out simply by a notation which shows that the subject of the adjective must be the same with the subject of the verb. This renders the derivational procedure of the standard theory useless: no need any longer to invoke a more basic structure of type "The moon shines and is happy" which is subject to a transformation.

Who are the inventors of HPSG ?

Ivan A. Sag (professor of linguistics and symbolic systems at Stanford University) and Carl Pollard (professor of linguistics at Ohio State University).

What is the utility of a computational implementation of a HPSG analysis?

The main utility consists in the fact that the analysis becomes testable and it thereby may offer hypotheses concerning the psychological plausibility of the way the structure subject to analysis is appropriated.

Why does HPSG reject Chomsky's concept of movement?

Because it discovered no convincing evidence that movement really does exist.

Are there any HPSG descriptions of Romanian?

Yes, there are. I would first mention Monachesi's analyses of weak pronouns. There is also an analysis of the multiple negation by Emil Ionescu (published in the Proceedings of Formal Grammar Conference, Utrecht 1999). Ana Maria Barbu proposed an analysis of the constituent order in the NP (the work is in print). Finally, a promising fact is that several dissertations of the students interested in HPSG applications exist.

How does HPSG treat Chomsky's concept of movement?

HPSG makes no use of any operation of movement whatsoever, because it does not find any convincing evidence that this operation really does exist as a part of the mental grammar.

How does HPSG treat the agreement phenomena?

Pollard and Sag ("Head-driven Phrase Structure Grammar", Chicago University Press, 1994, 73-88) treat agreement according to the elements involved in this relationship:

1. Pronoun-antecedent
2. Subject-verb
3. Determiner-noun

We must emphasize the fact that this classification does not follow from theory-internal reasons. That is, it is not determined by the fact that someone works in HPSG. This classification is determined by the nature of the language subject to investigation - here, English.

Please describe briefly how phrasal types are treated in HPSG.

The essential element in the HPSG treatment of phrasal types is the property of having a non-empty value for the feature DAUGHTERS. That is, a phrase is bound to have inner structure, reflected in the fact that it can be decomposed according to elements that may be either words or phrases (but not morphemes). This is the most general property of phrases. Further on, the diversity of phrasal types depends on the language subject to investigation. For instance, Romanian, but not also English, bears a phrase of type head-marker used to 'make obvious' certain direct objects NPs:

Ion o iubeste pe Maria

John loves Mary

How is lexical information organized within HPSG?

The main 'levels' of lexical information within HPSG are the grammatical one, the semantical one, the phonological one and the pragmatical one. There is also a level accounting for the word placement within a phrase.

What is specific to lexical representations in HPSG is the fact that they are rich - if compared for instance with GB lexical representations. This is because lexical representations account for facts - such as long distance dependencies, quantification or binding - which in other grammatical frameworks are treated as independent and autonomous phenomena.

II

SEMI-AUTOMATIC GENERATION OF WORDNET TYPE ROMANIAN SYNSETS AND CLUSTERS

On the Semiautomatic Generation of WordNet Type Synsets and Clusters with Special Reference to Romanian

Florentina Hristea

1 Introduction. What is WordNet

WordNet is a proposal for a more effective combination of traditional lexicographic information and modern high-speed computation. It is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets (*synsets*), each representing one underlying concept. Different relations link the synonym sets, WordNet being organized according to semantic relations which are indicated by pointers between synsets.

WordNet primarily represents *an interactive lexical data base* developed, during the last 15 years, at Princeton University by a group of researchers led by George Miller. At the same time, WordNet can be viewed as *a semantic dictionary* since words are located according to conceptual affinities with other words, unlike the case of classical dictionaries where words are ordered alphabetically. Although it resembles *a thesaurus*, WordNet is much more useful to Artificial Intelligence applications since it is enriched with an impressive set of relations among words and word meanings. WordNet distinguishes between semantic relations and lexical relations, but the emphasis is on semantic relations between meanings. Therefore, unlike standard dictionaries, WordNet organizes lexical information in terms of word meanings, rather than word forms. WordNet maps word forms in word senses using the syntactic category as a parameter. Thus, words belonging to the same syntactic category which can be used to express the same meaning are grouped into a single set, called *synset*. Therefore the "building block" of WordNet is *a synonym*

set (synset) of all words that express a given concept. Polysemous words belong to more than one synset. For instance, corresponding to the English word *computer*, two different meanings are defined in WordNet. It therefore belongs to two distinct synsets, as follows:

{computer,data processor,electronic computer,information processing system}

and

{calculator, reckoner, figurer, estimator, computer}.

In its most used version (ver.1.6), WordNet contains 129,509 English words organized in 99,643 synsets, with the network using a number of 229,152 nodes. Words and concepts are linked through a total of 299,711 semantic relations. However, all numbers are approximate since WordNet continues to grow. Version 1.7 is now accessible as well, at

<http://www.cogsci.princeton.edu/~wn/obtain/>

The most ambitious feature of WordNet is most probably the semantic attempt and, in this respect, WordNet resembles a thesaurus more than a dictionary. It equally represents *an on-line thesaurus* and *a semantic network*.

The rich set of semantic relations established among synsets is what makes this semantic network so powerful and useful for various types of applications. Examples of semantic relations existing in WordNet are **synonymy**, used in order to form synsets, **hypernymy** and **hyponymy**, corresponding to the *isa* relation and to *reverse isa* respectively, **meronymy**, corresponding to the *part of* relation, the **causal** relation referring to verbs and others. Using the *isa* relation nouns and verbs are structured in WordNet as *hierarchies*. Adjectives and adverbs are organized according to a different structure - *the cluster*. As its authors note [Miller et. al., 90], the advantage of imposing this syntactic categorization on WordNet "is

that fundamental differences in the semantic organization of these syntactic categories can be clearly seen and systematically exploited.” Nouns are organized in lexical memory as topical hierarchies, verbs are organized by a variety of entailment relations, and adjectives and adverbs are organized as N-dimensional hyperspaces. Additionally, the typical properties of a specific concept are stated as a *gloss* attached to each of the concepts. The gloss includes a definition, one or more supplementary explanations and one or more examples.

WordNet has been recognized as a valuable resource in the *human language technology* and *knowledge processing* communities.

The human language research community has encouraged the development of WordNets for languages other than English, at the same time concentrating on the possibility of automatically generating such huge lexical data bases. The main reason for this is the desire and the necessity to create a **uniform ontological infrastructure across languages** that will simplify **machine translation** from a language to another and will facilitate the use of the same **reasoning schemes** and **algorithms** developed in conjunction with the American WordNet.

2 The Translation Algorithm

The algorithm for translating a given English synset into the corresponding synset in a language other than English will be using so-called “elementary sets” or **e-sets**, a concept introduced in [Nikolov and Petrova, 00]. An e-set corresponds to a meaning of a word and can be defined as follows:

Definition 2.1

An *e-set* relative to a word is the set of synonyms corresponding to a specific meaning of that word.

Let us denote by EW any English word and by FW any foreign word, namely a word of a language other than English. Let **eword** of the fol-

lowing sequence be an EW, while *fword1*, *fword2* and *fword3* are its corresponding translation equivalents (according to the appropriate bilingual dictionary):

eword *fword1*; *fword2*, *fword3*

In order to distinguish among *fword1*, *fword2* and *fword3* two different separators are used in standard paper dictionaries. A semicolon separates different meanings of a given word. A comma separates synonyms which refer to one and the same meaning of the word. (In this case *fword2* and *fword3* are synonyms). This is the form of a bilingual dictionary which will be used by the programs implementing the proposed translation algorithm. In the above example the involved e-sets are

$\{fword1\}$ and $\{fword2, fword3\}$.

The computer programs which implement the translation algorithm will generate the list of all e-sets of FWs corresponding to the meaning of all EWs occurring in a given English synset. The foreign synset corresponding to the studied English one is formed of one or more of the generated e-sets (which can be adjoined). The "candidates" for inclusion in the foreign synset are *labeled e-sets*, namely those e-sets which contain *labeled words*.

In order to label the FWs belonging to the generated e-sets, we have decided to first label the EWs belonging to the English synset. These EWs will be labeled with integer numbers ranging from 1 to n (where n is the size of the synset, namely the number of words it contains), in the order of their occurrence. After labeling the EWs of the original synset, the FWs of the generated e-sets are looked up in the corresponding bilingual dictionary. Each time an EW of the given synset represents the translation, according to the dictionary, of a FW, the corresponding FW receives the label of that EW. If any word of a foreign e-set can be

translated into a word of the English synset using the bilingual dictionary, the whole foreign e-set is moved to the "list of candidates". As noted in [Nikolov and Petrova, 01], when completed, this list of candidates is the most important preliminary result. The appropriate foreign synset must be a compilation of some e-sets belonging to this list. Various *evaluating functions* which sort the extracted e-sets and outline the most adequate ones have been developed. In order to define such evaluating functions let us refer to the following concepts:

Definition 2.2

The *label of an e-set* represents the number of labels assigned to the words belonging to that e-set.

Definition 2.3

An e-set is *unlabeled* if it contains no labeled words.

Any word can have one or more labels assigned to it (as well as no label at all). The most common evaluating function which is proposed in the literature [Nikolov and Petrova, 01] takes as argument an e-set and has a value given by the very label of that e-set. A variant of this evaluating function is that which divides the number representing the label of the e-set to the size of the same e-set.

As far as we are concerned, we have taken into consideration the evaluation function which is defined below.

Each EW belonging to the given English synset will have a label (represented by an integer number from 1 to n , where n is the size of the synset) and the labeling of the FWs belonging to the e-sets is performed according to this label. The labels of the foreign words which differ from the label of the corresponding EW will be considered as representing two points, while the others represent just one point. The value of the evaluation function relative to a specific e-set is given by the total number of points corresponding to that e-set divided by its size.

Having defined all necessary concepts, one can now state the algorithm for generating the foreign e-sets corresponding to a given English synset:

Algorithm 2.1

Input: The file containing the English synsets and the two files representing the two bilingual dictionaries (for instance, the English-French and the French-English dictionary respectively).

- 1. Create (by consulting the appropriate bilingual dictionary) the e-sets corresponding to each word of the given English synset.
- 2. Label the English words belonging to the given English synset.
- 3. Label each of the e-sets generated at Step 1.
- 4. Remove all unlabeled e-sets.
- 5. Evaluate the e-sets (using the assigned labels and an evaluating function).

Output: The sorted list of e-sets corresponding to the given English synset.

The translations in the foreign language of the words occurring in the English synset are extracted from the bilingual dictionary as follows:

e-word1	meaning11; meaning12; ... ; meaning1m ₁
.....	
e-wordn	meaningn1; meaningn2 ; ... ; meaningnm _n

The set of e-sets generated by Algorithm 2.1 is of the following form:

$$\{\{\text{meaning}_{ij}\} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\}.$$

The *foreign synset* will be generated using this set.

In the automatic generation of the *foreign synset* corresponding to a given English synset we shall also take into account

Remark 2.1

Of all possible meanings of a word, only one refers to a specific concept (to which a synset corresponds).

Using the sorted list of e-sets generated by Algorithm 2.1 (namely the evaluated e-sets), the meaning (elementary set) evaluated with the highest value will be chosen corresponding to each English word. Let this meaning, corresponding to *ewordj*, be *meaningj_{i_j}*.

The *foreign synset* will be generated using the e-sets obtained by means of Algorithm 2.1, taking into account Remark 2.1 and according to

Algorithm 2.2

Input: The sorted list of e-sets generated by Algorithm 2.1 corresponding to the given English synset [*eword1*, *eword2*, ..., *ewordn*].

1. Compute the foreign synset as having the following form:

$$\{meaning1_{i_1}\} \cup \{meaning2_{i_2}\} \cup \dots \{meaningn_{i_n}\}, 1 \leq i_j \leq m_j,$$

$$\forall j = \overline{1, n}.$$

2. Delete words occurring in more than one e-set from this union, such that each word will occur just once.

Output: The foreign synset corresponding to the given English synset.

Algorithms 2.1 and 2.2 have been implemented in Prolog and tested by us, with very good results, in the case of *Romanian nouns*. All tests performed have been using the original WordNet 1.6 in its Prolog-readable format. In order to test the algorithms we have used fragments of bilingual

dictionaries since complete Romanian-English and English-Romanian dictionaries in electronic format were not available. We have randomly chosen a number of 200 English noun synsets for which we have automatically generated the corresponding Romanian ones. Since most English synsets contain two words, our data sample was chosen according to the same pattern. Thus, out of the 200 considered English synsets, 179 contained two English nouns, 4 synsets contained 3 English nouns and 17 synsets contained more than 3 English nouns (between 4 and 7 words). The number of e-sets involved in the experiment was of 616.

The fragments of electronic bilingual dictionaries used in the experiment were created by us, using [DRE, 73] and [DER, 74]. Our electronic bilingual dictionaries contained a total of 1164 words, out of which 278 were English nouns (corresponding to the 200 studied English synsets) and 886 words were the corresponding Romanian nouns representing their translations. The files containing these bilingual dictionaries are part of the input for the computer program implementing Algorithm 2.1. The Prolog format of the dictionaries, used in our Prolog implementation of all algorithms, can be seen in §3.2.2. The generated noun Romanian synsets corresponding to the 200 studied English ones and representing the output of Algorithm 2.2 can be seen on the web¹. The generated Romanian synsets were validated by Romanian linguists using the latest bilingual dictionaries and the corresponding gloss indicated in the American WordNet. As noted before, this gloss contains the explanation corresponding to a synonym string, thus containing the meronym, or the "mother" concept from a higher level in the hierarchy.

Actually, when testing the translation algorithm relatively to Romanian nouns, we have noticed that, in several cases, Algorithm 2.2 has generated more than one Romanian synset corresponding to the given English one. This was the case when Algorithm 2.1 had as output a list

¹<http://phobos.cs.unibuc.ro/oric/topic2.html>

of e-sets (corresponding to different meanings of the same word) that had been evaluated with the same value. Each such e-set then represented a candidate and led to a different Romanian, or, in general, foreign synset. In such cases the correct foreign synset will be chosen from the list of synsets generated by Algorithm 2.2 according to the gloss of the given English synset. The computer program implementing Algorithm 2.2 must therefore provide as output the gloss as well, since it is necessary in the validation performed by linguists.

When performing tests for Romanian nouns it turned out that for almost 96% of the considered English synsets the generated Romanian ones were correct. The remaining ones did not have correct Romanian counterparts mostly because of wrong or missing data in the bilingual dictionaries. We consider this result a very successful one, since it is well known that one can not work 100% automatically when dealing with linguistic resources.

Also in order to facilitate the experiment, when choosing our sample of English synsets a necessary step was that of removing the synsets with proper names, compounds and collocations. These should be dealt with separately and with a more significant contribution on the part of the linguists. However, the presented algorithms are sufficient for building a *core* of synsets corresponding to all four parts of speech in more or less any language other than English, providing that good bilingual dictionaries in electronic format exist for the specific foreign language involved.

The greatest advantage of the proposed translation algorithm is the ability to create synsets which may include foreign words that would not be extracted from the input resource at the first step of the work. Thus, even if a foreign word occurs in the English-Romanian dictionary, for instance, but is missing from the Romanian-English one, there is still a big chance for this word to be included in the final resulting synset. (The only necessary condition for this is the presence in the list of candidates of

an e-set which includes that word). This is a very important fact considering how incomplete bilingual dictionaries usually are. This algorithm, therefore, *does not represent a simple mirror translation*.

Obviously, when using Algorithms 2.1 and 2.2 for specific languages, various difficulties will occur according to what is typical of each language at morphological and derivational level. When testing a variant of the presented translation algorithm for *Bulgarian noun synsets*, for instance, phenomena like the lack of a regular conversion in Bulgarian, the translation of a gerund by a deverbial noun or by a special type of an infinitive or a subordinate clause, the existence of rich systems of participles and others are taken into account in [Nikolov and Petrova, 01]. Reported results are however also very good.

A general difficulty, of a different nature, encountered no matter what language is taken into consideration, consists of what we might call the cross-language wide meaning of a given word. Namely, a word in one language sometimes covers a relatively wide concept and is connected to more than one word in another language where each of the words it is linked to describes a more specific concept. This is a very important issue from the WordNet approach point of view, since in WordNet synsets exist according to the corresponding underlying concepts.

In spite of such difficulties, however, we consider the presented translation algorithm as being appropriate for performing a semiautomatic extraction of the *core* of a foreign WordNet from the original WordNet 1.6. In what follows, we shall establish how this general algorithm must be enriched in order for it to perform the semiautomatic generation of **adjective synsets and clusters** in languages other than English.

3 Generation of adjective synsets and clusters

3.1 Adjectives in WordNet

WordNet divides adjectives into two major classes : *descriptive* and *relational*. Chromatic *color adjectives* are regarded as a special case.

A *descriptive adjective* is one that ascribes a value of an attribute to a noun. That is to say, x is *Adj* presupposes that there is an attribute A such that $A(x) = Adj$. For instance, *low* and *high* are values for the attribute HEIGHT. WordNet contains pointers between descriptive adjectives and the noun synsets that refer to the appropriate attributes.

The *semantic organization* of descriptive adjectives is entirely different from that of nouns. The hyponymic relation that generates nominal hierarchies in the case of nouns is not available for adjectives. The semantic organization of adjectives is more naturally thought of as an abstract hyperspace of N dimensions rather than as a hierarchical tree. The basic semantic relation among descriptive adjectives is *antonymy*.

The importance of antonymy in the organization of descriptive adjectives is understandable when it is recognized that the function of these adjectives is to express values of attributes, and that nearly all attributes are bipolar. Antonymous adjectives express opposing values of an attribute. For example, the antonym of *heavy* is *light*, which expresses a value at the opposite pole of the WEIGHT attribute. In WordNet this binary opposition is represented by reciprocal labeled pointers: *heavy!*→*light* and *light!*→*heavy*. In the Prolog implementation of WordNet, which we have been using, the **ant** operator specifies antonymous words and all Prolog facts using this operator are included in the file **wn_ant.pl**. Descriptive adjectives that do not have direct antonyms are said to have indirect antonyms by virtue of their semantic similarity to adjectives that do have direct antonyms. A similarity pointer was used to indicate that the adjectives lacking antonyms are similar in meaning to adjectives that do have antonyms. In the Prolog implementation of WordNet the **sim** operator

specifies that two synsets are similar in meaning and all Prolog facts using this operator are included in the file **wn_sim.pl**.

Descriptive adjectives therefore ascribe to their head nouns values of (typically) bipolar attributes and consequently are organized in terms of *binary oppositions* (*antonymy*) and *similarity of meaning* (*synonymy*).

Gross, Fischer, and Miller (1989) proposed that adjective synsets be regarded as *cluster of adjectives* associated by semantic similarity to a focal adjective that relates the cluster to a contrasting cluster at the opposite pole of the attribute. Also Gross, Fisher and Miller distinguish direct antonyms like *heavy/light*, which are conceptual opposites that are also lexical pairs, from indirect antonyms, like *heavy/weightless*, which are conceptual opposites that are not lexically paired. Under this formulation, all descriptive adjectives have antonyms; those lacking direct antonyms have indirect antonyms, i.e. are synonyms of adjectives that have direct antonyms.

In WordNet, direct antonyms are represented by the antonymy pointer '!→'; indirect antonyms are inherited through similarity, which is indicated by the similarity pointer, '&→'. The configuration that results is illustrated in Figure 1 for the cluster of adjectives around the direct antonyms, *wet/dry*, which define the attribute WETNESS or MOISTNESS, an example often used by various authors. When analyzing this cluster of adjectives, one sees that *moist*, for instance, does not have a direct antonym, but its indirect antonym can be found via the path *moist&→wet!→dry*.

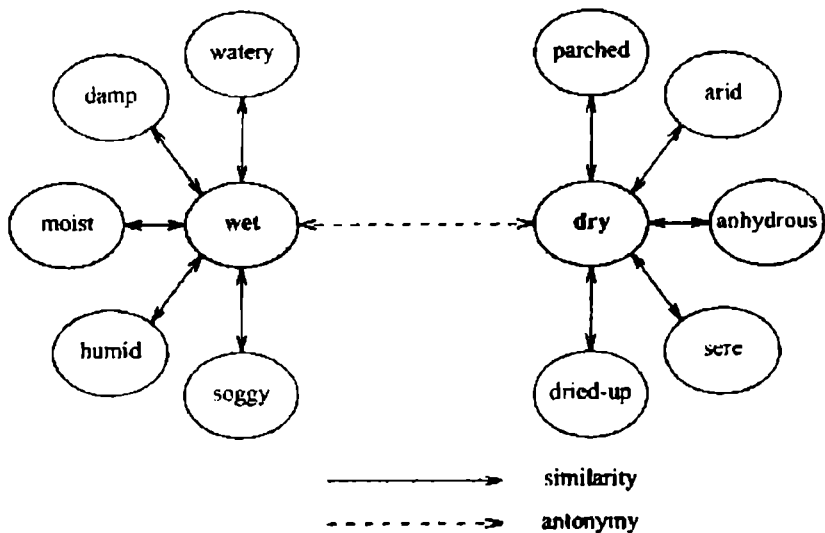


Figure 1. Bipolar Adjective Structure

Corresponding to Figure 1, which intuitively presents the structure of an adjective cluster in WordNet, one obtains the following bipolar cluster having as head the antonym pair *wet/dry*:

```
{ [WET, DRY, !] watery,& damp,& moist,& humid,& soggy,& }
{ watery, tearful, teary, wet, & }
{ damp, wet,& }
{ moist, wet,& }
{ humid, muggy, steamy, sticky, sultry, wet,& }
{ soggy, saturated, sodden, waterlogged, wet,& }
-
{ [DRY, WET, !] parched,& arid,& anhydrous,& sere,&
dried-up,& }
{ parched, dehydrated, desiccated, dry,& }
{ arid, waterless, dry,& }
{ anhydrous, dry,& ((chem) with all water removed)}
{ sere, shriveled, withered, wizened, dry,& (used of vegetation)}
```

{ dried-up, dry,& ("a dry water hole") } }.

One can notice that a cluster has two distinct parts. Each half of the cluster starts with a synset called *head synset*. The first two items of the head synset represent the antonym pair which defines the cluster and are capitalized. The antonym pair is followed by adjectives representing similarity pointers (watery,& damp,& etc.), one to each synset similar in meaning of the corresponding half of the cluster. Each such synset contains a reciprocal pointer for returning to the head synset. One also notices that the similarity pointers occurring in the head synset are, in fact, words occurring on the first position within the synsets related by similarity with the synset to which the adjective having a direct antonym belongs. The two distinct cluster parts are separated by a hyphen and the entire structure is enclosed between square brackets. Each bipolar cluster stands alone, and coding is restricted to within-cluster relations.

The significance of existing exceptions is not obvious and we believe, together with the authors of WordNet, that the presented model accounts for the great majority of the English descriptive adjectives. The importance of the similarity relation is obvious.

In WordNet semantic relations are represented by a pair of *synset_ids*, in which the first *synset_id* is generally the source of the relation and the second is the target.

A *synset_id* is a nine byte field in which the first byte defines the syntactic category of the synset and the remaining eight bytes are a *synset_offset*, indicating the byte offset in the file that corresponds to the syntactic category. In the Prolog version of the WordNet database, the *synset_ids* are used as unique synset identifiers. As it was already noted, in the Prolog implementation of WordNet the **sim** operator is used in order to designate the similarity relation, as in the example **sim**(302425348, 302425924). In

general,

sim(*synset_id*, *synset_id*).

is a Prolog fact specifying that the second synset is similar in meaning to the first synset. This means that the second synset is a satellite of the first synset, which is the cluster head. This relation only holds for adjective synsets contained in adjective clusters.

Relational adjectives in WordNet are assumed to be stylistic variants of modifying nouns and so are cross-referenced to the noun files. Relational adjectives, which were first discussed at length by Levi (1978), mean something like "of, relating/pertaining to, or associated with" some noun, and therefore play a role similar to that of a modifying noun (as in *atomic bomb*). Relational adjectives differ from descriptive adjectives in that they do not relate to an attribute. Therefore they do not refer to a property of their head nouns. Since relational adjectives do not have antonyms, they can not be incorporated into the clusters that characterize descriptive adjectives. WordNet maintains a separate file of relational adjectives with pointers to the corresponding nouns. Each synset consists of one or more relational adjectives, followed by a pointer to the appropriate noun.

In what follows, we shall be concerned with the semiautomatic generation of *adjective clusters* in languages other than English, and will therefore refer solely to descriptive adjectives, which can be organized as this type of structure.

3.2 Semiautomatic generation of adjective synsets

3.2.1 The algorithm

In order to translate English adjective synsets into a foreign language Algorithms 2.1 and 2.2 have been used. When translating from English to any other language the *id* which is associated to a synset is not modified. This means that the similarity relation existing between two English

synsets will be maintained after performing the translation and will occur among the foreign language adjective synsets as well.

A special problem is posed by synsets containing a single word. In this case it is impossible to tell which meaning of the word was involved in the creation of the specific synset if one has access to no additional information. The meaning can be guessed only from the gloss. However, in such cases we have used a strategy which consists in enriching the given synset with new adjectives that suggest the meaning of the one occurring in this synset. The new words are obtained using the similarity relation that typically exists in WordNet among adjective synsets. Thus, in order to enrich the given synset with new words, the adjectives occurring on the first position within synsets semantically linked to the original one via the similarity relation have been chosen. These words have been appended to the original synset, starting from the second position. This idea was inspired by the way in which adjective clusters are organized and structured.

The necessary list of e-sets in connection with the given English synset will be generated using Algorithm 2.1. When creating the foreign adjective synset representing the translation of the given English one, Algorithm 2.2 will combine all maximally evaluated e-sets corresponding to each of the words occurring in the English synset. In those cases when more than one e-set will be maximally evaluated corresponding to the same English word, Algorithm 2.2 will generate more than one foreign synset. The final decision concerning the correct translation is then made according to the gloss.

In order to illustrate how Algorithms 2.1 and 2.2 work in the case of adjective synsets let us consider the English synset having the *id* 302428719 and containing the unique adjective *sticky*. We shall perform the translation to Romanian of this synset. Let us note that the chosen target language is not essential for the point that we are trying to make here.

The presented results are the output of various Prolog programs.

Since the given English synset contains only one word, it will be enriched as mentioned, according to the similarity relation. After searching the database one comes to the conclusion that the only similarity relation (denoted by the **sim** operator) is

sim(302428719, 302425348).

as well as its symmetrical relation. The synset having *id* = 302425348 contains the unique adjective *wet*. The given English synset is therefore enriched with this adjective. The evaluated e-sets obtained corresponding to the enriched synset, when using the evaluation function mentioned in §2 for Algorithm 2.1, are the following:

evset (302428719, sticky, 1.0, [lipicios, cleios, vascos]).

evset (302428719, sticky, 1.0, [umed, cetos]).

evset (302428719, wet, 1.0, [umed, jilav, ud]).

evset (302428719, wet, 0.6666666666666666, [ploios, umed, igrasios]).

Here *evset* is an operator designating evaluated sets. The first field represents the synset *id*, the second is the ASCII text of the word as entered by the lexicographer, the third gives the value of the evaluation function and the last denotes the foreign evaluated set.

In this case the computer program implementing Algorithm 2.2 has the following output:

English synset: [sticky]

Gloss:

(moist as with undried perspiration and with clothing sticking to the body; "felt sticky and chilly at the same time")

Romanian synset: [[lipicios,cleios,vascos,umed,jilav,ud], [umed,cetos,jilav,ud]]

One notices that the output consists of two possible Romanian synsets. However, only one of them corresponds to the meaning of *sticky* which

refers to the underlying concept of the synset having *id* = 302428719. The correct foreign (in this case Romanian) synset can be easily chosen according to the corresponding gloss.

Such enrichment with additional words coming from synsets related via similarity with the original one is not always necessary. However, when performed, the chances of empty foreign adjective synsets being obtained (due to the generation uniquely of unlabeled e-sets) are considerably reduced. This operation might produce a slight shift in meaning with respect to the underlying concept of the original English synset. However, only similar concepts are denoted by the involved relation, typical for descriptive adjectives, a fact which determines us to recommend the described strategy. Both translation with and without enrichment can be performed, giving linguists the opportunity to compare and to choose among the proposed foreign synsets, when equally taking into consideration the corresponding gloss.

3.2.2 Prolog implementation

The WordNet database in a Prolog-readable format is contained in the files **wn_*.pl** and uses synset *ids* as unique synset identifiers. A separate file has been created for each WordNet relation giving the user the ability to load only those parts of this very large database that they are interested in. Each Prolog database file contains information corresponding to the synsets and word senses contained in the WordNet database. Each line of a file contains an operator that corresponds to a WordNet relation. All lines with the same *operator* value are stored in the file **wn_operator.pl**. The general format of a line in a Prolog database file is

operator(field1, . . . , fieldn).

containing no spaces and being terminated with a newline character.

An **s** operator is present for every word sense in WordNet, while the **g** operator specifies the gloss for a synset. File **wn_s.pl** contains all WordNet

synsets, while file **wn_g.pl** contains all glosses.

File **wn_s.pl** contains all WordNet synsets and is formed of Prolog facts having the general form

$$s(\textit{synset_id}, \textit{w_num}, \textit{'word'}, \textit{ss_type}, \textit{sense_number}, \textit{tag_state}).$$

where:

- *synset_id* represents the synset identifier;
- *w_num* is the word number within the synset;
- *'word'* is the ASCII text of the actual word to which the Prolog fact refers;
- *ss_type* is a one character code indicating the type of the synset which designates the part of speech of *'word'*. (In the case of adjectives this type can be *a*, standing for adjective, or *s*, standing for adjective-satellite);
- *sense_number* is a natural number denoting which sense of the word the synset refers to; sense ordering is performed according to the *tag_state*;
- *tag_state* can be 1 or 0, having the significance
 - 1 - the senses of the word are ordered according to frequency of use
 - 0 - the senses of the word are not ordered based on frequency of use,but
 - according to other criteria.

An *s* operator is present for every word sense in WordNet.

An example of such a Prolog fact, corresponding to the English adjective *abridged*, is the following:

$$s(300004481, 1, \textit{'abridged'}, a, 1, 0).$$

Due to the dimensions of the WordNet database, in order to decrease computer time, file **wn_s.pl** has been divided by us into smaller files containing all Prolog facts referring to the same part of speech. This file is part of the input of all programs dealing with the automatic generation of foreign synsets.

In order to decrease computer time even more, while eliminating all information which is unnecessary for our present task, file **wn.s.pl** containing only Prolog facts corresponding to adjectives has been processed even further. Computer programs have been written in order to obtain a simplified form of the Prolog facts referring to adjectives.

First of all, parameters *a* and/or *s*, denoting the part of speech, have been eliminated. The second and last two parameters have also been found unnecessary for our present purpose. Capital letters and special symbols like apostrophes have equally been eliminated, as well as Prolog facts containing proper nouns and collocations. The latter should be subject to a separate study. The resulting file containing adjective synsets includes Prolog facts of the following form:

s(300003469,'emergent').

s(300003469,'emerging').

s(300003469,'nascent').

Finally, the resulting file has been processed such that a unique Prolog fact was obtained corresponding to each synset. In the case of our present example, the following Prolog fact was created:

s(300003469, [emergent, emerging, nascent]).

Note that, when appending the words in order to create the synset, it is essential to maintain their order of occurrence due to our decision to always choose the adjective placed on the first position when enriching synsets containing a unique word via the similarity relation. The way in which adjective clusters are built also makes this necessary. The file containing Prolog facts of the above form, one fact corresponding to each adjective synset in WordNet, is part of the input of all computer programs implementing both Algorithm 2.1 and Algorithm 2.2.

We shall refer to this file as being "*the cleared, combined synset file*". Such cleared, combined synset files have been used both in the case of nouns and in the case of adjectives, when implementing the proposed algorithms. The present comments refer mainly to adjectives since processing adjective synsets required some additional steps beyond obtaining this type of file.

The cleared, combined synset file is part of the input of all computer programs implementing Algorithm 2.1 and Algorithm 2.2 when *translation without enrichment* is required.

Another important part of the input of all computer programs are the two bilingual dictionaries. In the case of the example we referred to in § 3.2.1 (the word *sticky*), a dictionary entry in the English-Romanian dictionary is of the form

sticky lipicios , cleios , vascos ; umed , cetos

We have been using only two separators, the comma and the semicolon. While a comma separates synonyms referring to the same meaning of the word, the semicolon is used as a separator between meanings. Each separator is preceded and followed by a blanc. The English word is the only one followed by more than one blanks. Corresponding to such a dictionary entry, the following Prolog fact has been obtained:

word1(sticky, [[lipicios, cleios, vascos], [umed, cetos]]).

The above predicate (*word1*) has two parameters, namely the English word and the list of corresponding e-sets. This list is included in square brackets. Each e-set is included in square brackets as well. The Romanian-English dictionary consists of similar Prolog facts.

On the web¹ you will find **the Java implementation** of all proposed algorithms, which uses similar formats of the bilingual dictionaries, as will be described there.

¹<http://phobos.cs.unibuc.ro/roric/topic2.html>

The next step in preparing the implementation of Algorithm 2.1 consists of automatically enriching those adjective synsets containing just one word by means of the similarity relation. The computer program which performs this operation receives as input the cleared, combined synset file and file **wn_sim.pl** of the WordNet Prolog database, which contains the Prolog facts corresponding to the existing similarity relations. The output is a file containing Prolog facts of the form

$$st1(synset_id, synset).$$

This file includes all adjective synsets in WordNet, with those having contained just one word being enriched. This is, in fact, **the final synset file**, representing the input for the computer programs implementing Algorithm 2.1 and Algorithm 2.2 when *translation with enrichment* is required.

As far as the Prolog implementation of semantic and lexical relations in WordNet is concerned, the issue is discussed in § 3.1 and § 3.3, when referring to the **sim** and to the **ant** operators respectively.

3.3 Semiautomatic generation of adjective clusters

The translation of English adjective clusters is completely ensured by the translation of the English adjective synsets and by that of the **ant** relation (denoting antonyms). Since the translation of adjective synsets has already been discussed in § 3.2, let us now refer to the translation of the **ant** relation. This becomes a very important issue when taking into account the fact that antonym dictionaries in electronic format do not exist for a great number of languages.

In the Prolog version of the WordNet database, which we have been using, semantic relations are represented by a pair of *synset_ids*, in which the first *synset_id* is generally the source of the relation and the second is the target, as is the case with the already mentioned **sim** operator. If two pairs *synset_id*, *w_num* are present, the operator represents a lexical

relation between word forms, where *w_num* specifies the word number for a specific word in a specific synset. If present, *w_num* indicates which word in the synset is being referred to. The **ant** operator, for instance, specifies antonymous words in the following form:

ant(*synset_id,w_num,synset_id,w_num*).

Thus, the significance of the following Prolog fact

ant(302425348, 1, 302429323, 1).

is that the first word of the synset having the *id* 302425348 and the first word of the synset having the *id* 302429323 are *direct antonyms*. This is a lexical relation that holds for all syntactic categories but is essential in the formation of adjective clusters. For each antonymous pair, both relations are listed (i.e. each *synset_id,w_num* pair is both a source and a target word).

When studying the contents of file **wn_ant.pl** of the WordNet Prolog database, which contains all Prolog facts referring to antonymous words, one easily notices that the great majority of these facts establish direct antonymy relations among words occurring as first elements within the synsets to which they belong. Less than 15 exceptions to this rule exist. These exceptions can be easily processed by a human operator retaining the new positions of the adjectives having direct antonyms. Under these circumstances, we have found it justifiable to formulate

Remark 3.1

The first word of an English adjective synset is the one possibly having a direct antonym.

Let us assume that all translated (foreign) adjective synsets exist and that they belong to a file named **wn_strans.pl**. Using Remark 3.1 and having generated file **wn_strans.pl** by applying the translation algorithm,

we can now formulate the algorithm for generating the foreign adjective clusters corresponding to the English ones:

Algorithm 3.1

Input: Files **wn_ant.pl**, **wn_sim.pl**, and **wn_strans.pl**

For each *synset pair* denoted by each Prolog fact of file **wn_ant.pl** perform steps 1. to 5.:

1. Look in file **wn_strans.pl** and find the foreign synsets representing the translations of the considered English ones.
2. Corresponding to each foreign synset found in **wn_strans.pl** at step 1. retain the first word of that synset. (This word pair will be used in the foreign cluster head).
3. For the same word pair look in file **wn_sim.pl** and take into consideration the **sim** clauses corresponding to each of the two synsets to which the two words of the cluster head belong.
4. Take into account all synsets denoted by the **sim** clauses chosen at step 3., synsets having the second *id* which occurs in the clause. Find the foreign synsets representing their translations in file **wn_strans.pl**.
5. Add each first word of these foreign synsets in the cluster head, together with the & pointer.
6. Add each "similar" foreign synset, ending it with the reciprocal similarity pointer.

Output: A file containing all foreign adjective clusters.

Algorithm 3.1 will generate foreign adjective clusters with a bipolar structure like the one described in § 3.1 and illustrated in Figure 1. Corresponding to this cluster, namely the one having the antonym pair [WET,

DRY, !] and [DRY, WET, !] respectively in the head synset, the following Romanian cluster has been generated:

{ [UMED, USCAT, !] inrourat,& stropit,& vascos,& umed,& umed,& cetos,& lipicios,& ploios,& lipicios,& umed,& }

{ inrourat, stropit, umezit, umed,& }

{ stropit, smaltat, umed,& }

{ vascos, cleios, lipicios, umed,& }

{ umed, igrasios, jilav, ud, umed,& }

{ umed, jilav, ud, umed,& }

{ cetos, aburit, umed, jilav, ud, umed,& }

{ lipicios, cleios, vascos, umed,& }

{ ploios, umed,& }

{ lipicios, cleios, vascos, umed, jilav, ud, umed,& }

{ umed, jilav, ud, umed,& }

-
{ [USCAT, UMED, !] arid,& uscat,& secat,& uscat,& uscat,& uscat,& }

{ arid, uscat, sec, uscat,& }

{ uscat, arid, sterp, sec, uscat,& }

{ secat, uscat,& }

{ uscat, ofilit, vestejit, zbarcit, uscat,& }

{ uscat, arid, sterp, sec, uscat,& }

{ uscat, arid, insetat, uscat,& }]

Note that not all existing similarity relations have been used in this case, since our aim was only that of offering an example of the *type* of structure which is generated by Algorithm 3.1.

Our Romanian cluster only illustrates the basic coding devices used in the American WordNet, which we have also made use of. At this early stage of our study we have been concerned uniquely with creating the *WordNet type* cluster structure and have not tried to distinguish among different subsenses or different privileges of occurrence. We have equally not tried to indicate the limitation of certain adjectives as to the syntactic positions they can occupy, a word-form limitation which in WordNet is coded for individual adjectives. This can easily be achieved once the basic algorithm has been established. Other issues, such as the capitalized pointers sometimes occurring in the structure of WordNet clusters, which serve as "see also" cross-references to related clusters, have also been ignored for the time being. All these and others represent topics for future study.

Obviously, according to the chosen target language, various difficulties of linguistic nature will be encountered. For instance, identical foreign synsets might be generated by Algorithm 3.1 corresponding to different English ones, namely to different meanings and concepts. This is the case when an English polysemous adjective will have one or more meanings in English that do not exist in the target language, a situation which is called *semantic loan*, leading to *loan translation*. Linguistic validation of the output of computer programs implementing Algorithm 3.1, or any other algorithm of the same type, for that matter, will always be necessary. However, we consider that Algorithm 3.1 accounts for the great majority of cases when dealing with adjective clusters of WordNet type.

4 Final Remarks

WordNet has been recognized as a valuable resource in the human lan-

guage technology and knowledge processing communities. Many researchers who use WordNet especially in the field of Artificial Intelligence view it primarily as a lexical *knowledge base* and make subsequent use of it. Knowledge processing has gained new dimensions in the U.S. due to the existence of WordNet. Its applicability has been cited in more than 200 papers and systems have been implemented using it. Many groups of researchers expressed their interest in WordNet applications in various fields, such as : Information Retrieval, Information Extraction, Word Sense Disambiguation, Text Inference, Natural Language Generation, Learning, Knowledge Acquisition and others.

The human language research community has encouraged the development of WordNets for languages other than English, at the same time concentrating on the possibility of automatically generating such huge lexical data bases. The main reason for this is the desire and the necessity to create *a uniform ontological infrastructure across languages* that will simplify translation from a language to another and will facilitate the use of the same reasoning schemes and algorithms already developed in conjunction with the American WordNet.

Acknowledgments

1. The author would like to thank Prof. Dr. Theodor Hristea of the Faculty of Letters, the University of Bucharest, for having provided constant linguistic guidance concerning WordNet, as well as for the validation of Romanian generated synsets and clusters.
2. The author would like to thank Master of Science students Cristina Vață, Claudia Burtea and Mihai Sima of the Faculty of Mathematics, the University of Bucharest, as well as Raluca Elena Gălățanu of the Faculty of Letters of the same university, for having assisted the RORIC-LING team, with much dedication, during this second phase of the BALRIC-LING project.

References

- [Fellbaum, 98] Fellbaum, C. (Ed.): "WordNet: An Electronic Lexical Database"; The MIT Press, Cambridge/London/England (1998).
- [Harabagiu, 99] Harabagiu, S.: "Lexical Acquisition for a Romanian WordNet"; Proc. EUROLAN '99, Iași, Romania (1999).
- [Miller et. al., 90] Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: "Introduction to WordNet: an on-line lexical database"; International Journal of Lexicography, 3,4(1990), 235-244.
- [Nikolov and Petrova, 00] Nikolov, T., Petrova, K.: "Building and Evaluating a Core of Bulgarian WordNet for Nouns"; OntoLex '2000 Report, Sozopol, Bulgaria (2000).
- [Nikolov and Petrova, 01] Nikolov, T., Petrova, K.: "Towards Building Bulgarian WordNet"; Proc. RANLP'01, INCOMA Ltd., Tzigov Chark, Bulgaria (2001), 199-203.
- [DRE, 73] Levițchi, L., "Dicționar român-englez" (ed.a III-a); Editura Științifică, București (1973).
- [DER, 74] "Dicționar englez-român"; Editura Academiei R.S.R., București (1974).

Appendix 1

GenSynsets - the Java implementation of the algorithms

George Ungureanu

GenSynsets is a tool conceived in order to facilitate the development of *WordNets* for languages other than English. It implements the algorithms described in the paper "On the Semiautomatic Generation of WordNet Type Synsets and Clusters with Special Reference to Romanian". **GenSynsets** can be used with reference to any foreign language for which you have bilingual dictionaries in electronic format. The program has been tested for the Romanian language and the corresponding output (an XML file) can be explored on the Web.

Installation

GenSynsets is written in the Java programming language and runs on the Java 2 platforms.

Requirements:

1. In order to run, **GenSynsets** requires the Java 2. Therefore the operating system of the computer on which **GenSynsets** is to be installed must be one for which an implementation of Java 2 exists (Windows 95/98/ME/NT/000, most Unix versions, MacOS X).
2. The system on which **GenSynsets** is to be installed must be sufficiently powerful (processor speed, memory). In the case of a PC, a minimum of 133 MHz and 32M RAM are necessary.

Installation:

1. Install Java 2 on your system. If Java 2 is already installed, skip this step. Corresponding to Windows, Linux, Solaris platforms, the necessary installation kits can be found at java.sun.com. The entire JDK development kit or only the JRE runtime environment can be installed. Using version 1.3 or a more recent version is recommended.
2. Make sure that the PATH variable contains the path to the java executable. Corresponding to Windows 95/98/NT this can be achieved by including in the `autoexec.bat` file a line of the following form:

```
SET PATH=c:\path; %PATH%
```

where `c:\path` is to be replaced with the actual path leading to the directory where the `java` executable is placed. Following this operation (and restarting your computer), as a result of the command

`java -version`

the answer of your system should look like:

```
C:\>java -version
java version "1.3.0"
Java(TM) 2 Runtime
Environment, Standard Edition
(build 1.3.0-C)
Java HotSpot(TM) Client VM
(build 1.3.0-C, mixed mode)

C:\>
```

no matter which directory the command was issued from.

3. Unzip the archive `gensynsets.zip` and place its content wherever you wish within the existing directory structure.

Remark: The above directions have concentrated especially on installation under the Windows operating system. In order to install and to run **GenSynsets** on any system, in essence, you must be able to install the Java 2 environment on that system, to unzip the archive `gensynsets.zip`, and then to run the (Java) class `GenSynsets`.

Usage

GenSynsets is designed to be used from the command line. The general form of the program command line is:

```
java -classpath .;jwordnet.jar GenSynsets -pos
noun|adj [-enrich] [-cs charset] [-l SynList]
WnDictPath E_F F_E OutFile
```

where:

- **-pos** specifies the part of speech to be considered : `noun` or `adj`
- **-enrich** is used only in the case of adjectives. Its occurrence triggers using the enrichment technique (see the paper).

- **-cs** sets the character set as it will appear in the output XML file (<?xml version="1.0" encoding="charset">). The default value is **iso-8859-1**.
- **-l SynList**; if this switch is present, foreign synsets will be generated only for English synsets whose offset is present in the **SynList**. If the switch is not present, the whole list of the English synsets (corresponding to the specified part of speech) is processed by the program.
- **WnDictPath** is the path to the WordNet Databases (c:\wn16\dict).
- **E_F**, **F_E** represent the English-Foreign (language), Foreign-English dictionaries, respectively.
- **OutFile** will contain the result of the processing (the corresponding foreign language synsets).

The files format

bilingual dictionaries

- the English-Foreign dictionary contains lines each of which includes the English word followed by a space and the corresponding translation equivalents:

```
eword fword1,fword2,fword3;fword4,fword5
```

In order to distinguish among **fword1**, **fword2**, etc. two different separators are used. A semicolon separates different meanings of the given word (**eword**). A comma separates synonyms which refer to one and the same meaning of the word (**eword**).

- the Foreign-English dictionary contains lines each of which includes the foreign word followed by a space and the corresponding translation equivalents:

```
fword eword1;eword2,eword3;eword4,eword5
```

In order to distinguish among **eword1**, **eword2**, etc. two different separators are used. A semicolon separates different meanings of the given word (**fword**). A comma separates synonyms which refer to one and the same meaning of the word (**fword**).

the output file

- The output is delivered as an XML file. Thus, the XML files produced by **GenSynsets** can be easily transformed, by means of

XSLT, into other formats (XML, HTML, etc.) and can be used by other applications. For more details see the DTD file (fsynsets.dtd) on which the output XML file is based.

Appendix 2. Sample output for Romanian nouns

English synset: {banishment, proscription}

the act of banishing someone

e-sets:

eword	e-set	score
proscription	{surghiunire, exilare}	2.0
banishment	{exilare, surghiunire, exil, surghiun, expulzare, ostracizare}	0.8333333

proposed foreign synset(s):

- {exilare, surghiunire, exil, surghiun, expulzare, ostracizare}

English synset: {ostracism}

the act of excluding someone from society by general consent

e-sets:

eword	e-set	score
ostracism	{ostracism}	1.0
ostracism	{ostracizare, surghiunire}	0.5

proposed foreign synset(s):

- {ostracism}

English synset: {substance, matter}

that which has mass and occupies space; "an atom is the smallest indivisible unit of matter"

e-sets:

eword	e-set	score
substance	{materie, substanta}	3.0
matter	{materie, substanta}	3.0
matter	{fond}	3.0
substance	{esenta, fond}	2.0
matter	{chestiune, problema}	1.0
matter	{lucru, fapt, obiect}	1.0
matter	{subiect}	1.0

proposed foreign synset(s):

- {materie, substanta}
- {materie, substanta, fond}

English synset: {gesture}

motion of hands or body to emphasize or help to express a thought or feeling

e-sets:

eword	e-set	score
gesture	{gest}	1.0
gesture	{gest, atitudine}	0.5

proposed foreign synset(s):

- {gest}

English synset: {universe, existence, nature, creation, world, cosmos, macrocosm}

everything that exists anywhere; "they study the evolution of the universe"; "the biggest tree in existence"

e-sets:

Eword	e-set	score
World	{lume}	7.0
Cosmos	{lume}	7.0
Creation	{lume, univers, natura}	5.3333335
Universe	{univers}	5.0
world	{univers, cosmos}	4.5
cosmos	{cosmos, univers}	4.0
macrocosm	{macrocosm, univers}	3.5
universe	{cosmos}	3.0
nature	{fire}	3.0
nature	{natura, fire}	2.0
nature	{character, temperament, fire}	1.3333334
nature	{natura}	1.0
nature	{esenta, character}	1.0
creation	{creatiune}	1.0
creation	{creatie}	1.0
existence	{existenta, fiintare, vietuire}	0.6666667
creation	{creare, faurire, zamislire}	0.6666667
existence	{prezenta, existenta}	0.5
existence	{fiinta, vietuitoare}	0.5
nature	{natura, specie, fel, gen}	0.5

proposed foreign synset(s):

- {univers, existenta, fiintare, vietuire, fire, lume, natura, macrocosm}

English synset: {advocate, proponent, exponent}

a person who pleads for a cause or idea

e-sets:

eword	e-set	score
advocate	{avocat}	1.0
exponent	{exponent}	1.0
proponent	{sustinator, partizan, adept}	0.6666667
advocate	{aderent, adept, partizan}	0.33333334
exponent	{interpret, indrumator, talmacitor}	0.33333334
exponent	{exponent, factor, reprezentant}	0.33333334

proposed foreign synset(s):

- {avocat, sustinator, partizan, adept, exponent}

English synset: {absolutism, tyranny, despotism}

dominance through threat of punishment and violence

e-sets:

eword	e-set	score
despotism	{despotism}	5.0
tyranny	{tiranie}	3.0
absolutism	{absolutism}	1.0

proposed foreign synset(s):

- {absolutism, tiranie, despotism}

Appendix 3. Sample output for Romanian adjectives

Adjective synsets obtained without enrichment

English synset: {brumous, foggy, hazy, misty}

filled or abounding with fog or mist; "a brumous October morning"

e-sets:

eword	e-set	score
foggy	{cetos, nebulos, neclar, confuz}	3.25
foggy	{cetos, neguros}	3.0
hazy	{incetosat, neguros}	3.0
brumous	{posomorat, cetos, neguros}	2.6666667
misty	{cetos, incetosat, innorat, aburit}	2.0
misty	{vag, neclar, confuz, neinteligibil}	1.5
hazy	{vag, neclar, estompat, sters}	0.5
hazy	{confuz, intunecat, nesigur}	0.33333334

proposed foreign synset(s):

- {posomorat, cetos, neguros, nebulos, neclar, confuz, incetosat, innorat, aburit}

English synset: {cloud-covered, clouded, overcast, sunless}

filled or abounding with clouds

e-sets:

eword	e-set	score
cloud-covered	{innorat, noros}	4.0
clouded	{innorat}	3.0
clouded	{innorat, posomorat}	1.5

proposed foreign synset(s):

- {innorat, noros}
-

English synset: {fair}

free of clouds or rain; "today will be fair and warm"

e-sets:

eword	e-set	score
fair	{bun, frumos}	1.0
fair	{ieftin}	1.0
fair	{balan, balai, blond, deschis}	0.75
fair	{frumos, curat, ingrijit}	0.6666667
fair	{drept, nepartinitor, impartial}	0.6666667
fair	{bun, frumos, placut, prielnic, favorabil}	0.6
fair	{cinstit, onest}	0.5
fair	{cinstit, deschis}	0.5
fair	{convenabil, acceptabil, accesibil, rezonabil}	0.5
fair	{bun, natural, firesc}	0.33333334
fair	{frumos, minunat, atragator, dragut}	0.25

proposed foreign synset(s):

- {bun, frumos}
- {ieftin}

Adjective synset obtained with enrichment

English synset: {fair}

free of clouds or rain; "today will be fair and warm"

e-sets:

eword	e-set	score
fair	{senin}	2.0
fair	{limpede}	2.0
clear	{clar, curat, luminos, limpede, senin}	1.4
fair	{frumos, curat, ingrijit}	1.3333334
fair	{citet, clar}	1.0
fair	{bun, frumos}	1.0
fair	{ieftin}	1.0
clear	{limpede, lamurit, inteligibil, clar, deslusit}	1.0
clear	{clar, perceptibil, lamurit, limpede, deslusit}	0.8
fair	{balan, balai, blond, deschis}	0.75
clear	{curat, neincarcata, negrevat, integ}	0.75
fair	{drept, nepartinitor, impartial}	0.6666667
fair	{bun, frumos, placut, prielnic, favorabil}	0.6
fair	{cinstit, onest}	0.5
fair	{cinstit, deschis}	0.5
fair	{convenabil, acceptabil, accesibil, rezonabil}	0.5
clear	{liber, deschis}	0.5
clear	{clar, patrunzator}	0.5
fair	{bun, natural, firesc}	0.33333334
fair	{frumos, minunat, atragator, dragut}	0.25

proposed foreign synset(s):

- {senin, clar, curat, luminos, limpede}

Some Linguistic Comments Concerning the Obtained Output

Theodor Hristea

We would like to mention, from the very beginning, that, in most cases, the computer programs implementing the proposed WordNet algorithms work correctly, and that, when the obtained results are not the best possible ones, it is mainly because of the imperfection of the existing bilingual dictionaries. The project web page shows primarily those situations in which the programs make mistakes, or in which they propose more than one Romanian synset, leaving it up to the linguist to choose the most adequate one, mainly according to the gloss. In what follows, we shall try to comment on the main types of mistakes which can occur as a result of automatic processing, and to briefly analyze the causes of these mistakes.

We would like to point out especially the following three types of situations: those in which the program has generated more than one Romanian synset, out of which one is correct, those in which no Romanian synset has been generated, and those, very rare cases, in which one or more synsets have been generated, none of them being correct.

In those cases when two or more Romanian synsets have been generated, among which the correct one occurs, finding it according to the gloss was generally an obvious operation for the linguist.

We consider as being much more interesting those situations in which no Romanian synset was generated. Most frequently, the cause for this is the imperfection of the bilingual dictionaries, which simply do not include those words. Sometimes only one of the dictionaries is to blame, usually the Romanian-English one, relatively poor concerning the number of entries, but also as far as the number of English words taken into consideration for performing translations is concerned. Due to this fact, there are many cases in which only unlabeled e-sets are obtained via the proposed algorithm. No Romanian synset will be generated in such cases.

Situations of different natures in which no Romanian synset is generated therefore exist. Either the word was not found in the English - Romanian dictionary, which directly affects the translation of English synsets containing a unique word, that are frequent enough in WordNet, or it was found but, corresponding to it, only unlabeled e-sets were generated. The latter situation is the most frequent. It is, for instance, the case of **crook**, having the meaning "a long staff with one end being

hook shaped", or the case of **wreckage**, having the meaning "the remains of something that has been wrecked".

Sometimes the Romanian synset generated by the program is incorrect because of the evaluation function which was implemented. Other evaluation functions should be implemented and tested in future studies. Most frequently, however, the evaluation function taken into consideration now does not work correctly again because of the incompleteness of the existing bilingual dictionaries. It is, for instance, the case of the synset formed with the unique word **rule** having the meaning "directions that define the way a game or sport is to be conducted", translated into Romanian by [**rigla**], as well as of the synset [**convention**], having the meaning, coming from diplomacy, "an international agreement". It was translated into Romanian by the synset [**adunare, intrunire, congres**], denoting the concept of "congress", instead of the correct [**conventie, acord, contract, inoiala, intelegere, pact, tratat**].

As we have already mentioned, the situation in which a Romanian word occurring in the English-Romanian dictionary is not found in the Romanian-English one is quite frequent. It is especially the case of nouns coming from verbs and having the significance "the action of...". Important and frequent Romanian words like **organizare** (coming from "a organiza" - "to organize") or **respingere** (coming from "a respinge" - "to reject"), occur as translations of various English words but are not to be found in the Romanian-English dictionary. This can determine the algorithm for the evaluation of e-sets to fail, since the absence of a word from the Romanian-English dictionary leads to a lower value of the corresponding e-set.

Also due to the incompleteness of existing bilingual dictionaries many recent borrowings which exist in Romanian (especially in mass-media) will not occur in the generated Romanian synsets.

In those, more interesting, cases in which the Romanian-English dictionary is not to blame, the cause of the errors which the programs generate is of a completely different nature. One should look for it in connection with concepts. In this case one must take into account the fact that English in general and American English, to which WordNet refers, in particular, is a much richer language than Romanian. Statistically speaking, while Romanian has a maximum of 150,000 words, American English includes approximately 450,000 words (according to information provided by the lexicographer St. Berg Flexner). But, in comparison with Romanian, English is a much more advanced language not only from a grammatical and lexical point of view. Quantitatively it includes more words or lexical units. However, English is much more advanced from the semantic point of view as well, since an English word often has a much richer semantic content than the corresponding Romanian one. Numerous words existing both in English and in Romanian are more polysemous in English than in Romanian. In other words, the polysemy of many English words is greatly superior to that of the corresponding

Romanian ones. For instance, the English word **feature** having the meaning of "an article of merchandise that is displayed or advertised more than other articles" has no correspondent in Romanian. No single word with this meaning exists. We are therefore obliged to perform translation using a group of words (a gloss), while the English synset containing the sole word **feature** which refers to this concept will have no Romanian counterpart. In this case the computer program did not work correctly. It is, once again, a situation which affects primarily English synsets containing a single word. Another example of an English polysemous word is **foundation**, which attracted our attention through one of its meanings, that of "a woman's undergarment worn to give shape to the contours of the body". This meaning of **foundation** does not exist in Romanian. The concept to which the synset containing the unique word **foundation** with this meaning refers to should be explained in Romanian by means of a gloss. No corresponding Romanian synset should exist. The computer program has again failed in this case, just as it has in the case of the English **quiver** having the meaning "a case for holding arrows".

Another situation in which the program did not work correctly refers to certain English nouns used with a negation. This is, for instance, the case of **matter** with negation, as in "they were friends and it was no matter who won the game". This English noun should be translated into Romanian by a collocation, centered around a noun which does not occur in the English-Romanian dictionary among the possible translations of **matter**. Another possibility is that it does occur, however by means of an equivalent of collocational type, that will not be used by the algorithm which the program implements. In such cases the program can not determine the Romanian (or, in general, the foreign) synset correctly. Specifically, in the case of **matter** used with a negation, several possible Romanian synsets have been generated. None of them is, however, correct, since none of them includes the noun **importanta** (importance), which occurs in the Romanian collocation corresponding to this meaning. This Romanian collocation represents a loan translation of the French "avoir de l'importance". Loan translations after French are extremely frequent in Romanian. This is why we feel the need for future programs to take into account collocations, both in English and in Romanian, or, more generally, in the target language.

Other times, the unique English noun of a synset is not translated into Romanian by a collocation but by a word having exactly the same form. Even so, the program does not work correctly in some of these situations. It is, for instance, the case of the English synset [**act**] which denotes the concept "lack of sincerity". It has been wrongly translated into Romanian by the synset [**fapta, fapt, act, actiune**] which contains, among others, a Romanian word having the same form - **act**. But this meaning of the English **act** - lack of sincerity - does not exist in Romanian. This represents an example of what linguists call "false friends". In such cases one deals with English words which exist in an identical or very close form in other languages as well, however without having the typical English meaning. It is also the case of the synset [**pattern**] having the non-existent meaning in Romanian "the

path that is prescribed for an airplane that is preparing to land at an airport" or that of the synset [cosmos] having the meaning "any of various mostly Mexican herbs of the genus Cosmos". Many of these meanings are typical to American English. Another example is offered by the synset [circumstances] denoting the concept "the state (usually personal) with regard to wealth" wrongly translated by the Romanian synset [imprejurari, circumstante, conditii]. This meaning of **circumstances** (plural) exists both in British and in American English, but not in Romanian.

Another source of difficulties was represented by nouns in plural form. Some of the English synsets contain nouns in singular form which should be translated by plurals in Romanian. Examples from this category are **foundation** translated by the plural **fonduri**, or **knowledge** translated by **cunostinte**. In order to deal with such situations we have decided to include the plural forms of these nouns in the Romanian-English dictionary which was used by the computer program. The program was thus able to take into consideration e-sets containing nouns in plural form as well.

In Romanian, as in other languages, like French, for instance, the relationship between homonymy and polysemy represents an extremely complicated issue, a problem which is not yet solved. In many cases, according to various researchers, one deals with two, three or even more homonymous words, while according to others with a unique polysemous word, having two, three or even more fundamental meanings, which are more or less related to one other. An example would be the word **bun** (**good**), which in Romanian is primarily an adjective having seven fundamental meanings. Secondly it represents a noun having two different plural forms, which are semantically specialized. The Romanian noun **bun** (**good**) having the plural **bunuri** has four meanings, while the same noun **bun** with plural form **buni** has only one meaning, that of grandfather. These situations occur quite frequently in Romanian. The computer programs designed within the framework of this project will produce better results when using dictionaries which treat possible homonyms, especially the so-called semantic ones, as a single polysemous word. Otherwise the gloss should be taken into account from the very beginning in order to establish the meaning, namely the concept to which the English synset refers.

To conclude, one can say that the main difficulties which occurred when automatically translating the English synsets into Romanian ones were generated by the so-called "false friends", by collocations, by loan translation, and by the fact that the polysemy of many English words is greatly superior to that of the corresponding Romanian words. At the same time, one must notice that most problems occurred when translating English synsets that contain a single word, the algorithm often being unable to decide among meanings. Such synsets should probably be subject to further investigation. On the other hand, we would like to emphasize the fact that, in the absence of truly competitive tools (with reference to

paper and electronic dictionaries) the realistic evaluation of the computer programs becomes rather difficult, if not almost impossible.

One can not conclude without pointing out some of the merits of the proposed algorithms. Let us start by noting, for instance, that, in spite of all mentioned difficulties, a great number of English synsets containing a unique polysemous word have been correctly translated into Romanian synsets containing a single polysemous word, as well. Examples are: the synset [**art**], correctly translated into [**arta**], or the synset [**creation**], again correctly translated into [**creatie**].

As it is well known, concepts are language dependent. In many cases it may happen that an English word covers a very wide concept, being linked to several Romanian words which refer to various related and much more specific concepts. One of the examples given for Bulgarian by Nikolov and Petrova (2001), concerning this aspect, stands for Romanian as well. It involves the synset containing the unique word **castle**. When translating **castle** into Romanian, words like **fortareata** (**fortress**) or **citadela** (**citadel**) will occur. They denote related but different concepts. We would like to carry this comment further, by noticing that this represents a situation when the proposed algorithm works correctly, by producing unlabeled e-sets which will then be rejected. The translation into Romanian of synset [**castle**] having the meaning "a large building formerly occupied by a ruler and fortified against attack" is a correct one.

Finally, one should notice the fact that, in most cases when the bilingual dictionaries were correct and complete, the implemented algorithm proved to work surprisingly well. Thus, in the case of concepts which are very close to one another in English, the existing subtle difference in meaning has been sensed by the algorithm which correctly maintains it in the Romanian translation. It is, for instance, the case of the English synsets [**banishment**, **proscription**] having the meaning "the act of banishing someone" and [**ostracism**] having the meaning "the act of excluding someone from society by general consent" respectively. The first was translated into Romanian by the synset [**exilare**, **surghiunire**, **exil**, **surghiun**, **expulzare**, **ostracizare**], while the second one was translated into the unique [**ostracism**]. The Romanian **ostracism** is the only of all these synonym words which also refers to consensus in making the banishment decision. Its occurrence in the second synset, as a unique element, points out the subtle difference between the two concepts to which the English synsets refer.

We would like to conclude by saying that such a study concerning the possibility of automatically or semiautomatically generating foreign synsets by starting from the American ones is undoubtedly useful, and seems promising enough. We encourage its continuation in the case of the Romanian language, and we suggest its enlargement due to the study of collocations in the near future. In order to perform a more or less complete study, these should be taken into consideration both in English and in Romanian, or, more generally, in the target language.

The RORIC-LING Bulletin

months 7 - 12

A number of **85** questions have been asked, by subscribers coming mostly from Romanian academic units, but not only. Software companies, both from Romania and from abroad, are also represented. Some of these questions have been asked more than once, as will be specified in the bulletin. There are four main categories of questions, corresponding to the topics of the BALRIC-LING Romanian part of the project, as follows:

- general questions;
- questions concerning the program GenSynsets;
- questions concerning the RORIC-LING implementation of WN algorithms;
- general questions concerning WordNet.

The questions have been grouped according to the topic they refer to (and not according to the subscribing date, namely not in chronological order). All information concerning the names and personal data of subscribers is stored in the RORIC-LING files but has been deleted from the bulletin, in order to facilitate reading and using this material, as well as the search process according to topic.

Some statistics:

Questions: 85

Country:	Romania	Other*
Questions:	67	18

Native Language:	Romanian	Other
Questions:	80	5

Activity Field:	Education	Research	Software Industry	Other
Questions:	41	16	16	12

* AUSTRALIA, AUSTRIA, FRANCE, GERMANY, ITALY, UNITED KINGDOM, UNITED STATES

General Questions

Could you recommend me some relevant publications on WordNet?

Those unfamiliar with WordNet should read "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

If already familiar with WordNet, we recommend the book "WordNet: An Electronic Lexical Database", which is now available from MIT Press. The book includes articles describing the design and contents of WordNet (an update to Five Papers on WordNet as well as papers reporting on research done with WordNet in the areas of linguistics, information retrieval, word sense disambiguation, semantic concordance building, text analysis, and knowledge engineering). The book and CD-ROM can be purchased directly from MIT Press.

Where can I find professional XML documentation in Romanian?

We do not know of any free XML documentation in Romanian on the web. However, a very good book, translated into Romanian, exists:

Lee Anne Phillips, XML. Teora Publishing House, 2001

I've downloaded the WordNet PC package but don't know how to install it. Can you help me, please?

You should have downloaded a file called "wn16pc.exe". If you downloaded it correctly, you should be able to simply double-click on this file and it should extract itself. Then you must follow the instructions in the INSTALL.txt file to actually install the WordNet package.

Where can I obtain WordNet manuals?

Please look at <http://www.cogsci.princeton.edu/~wn/doc.shtml>. You should see a list of WordNet manuals, available there online.

What are some WordNet-related projects?

Some WordNet-related projects are the following:

- Semantic networks - in languages other than English
- Web Interfaces - access WordNet over a network
- Local Interfaces - require files to be downloaded
 - .NET
 - C++
 - COM
 - dBase/MySQL
 - Java
 - Lisp
 - Palm
 - Perl
 - Prolog
 - Python
- Extensions - expand WordNet's features or integrate it into larger systems

More information on these projects can be found at
<http://www.cogsci.princeton.edu/~wn/links.shtml>

What is the Global WordNet Association?

The Global WordNet Association is a free, public and non-commercial organization that provides a platform for discussing, sharing and connecting wordnets for all languages in the world.

Questions Concerning the Program GenSynsets

Please explain why the final output is generated in XML format.

The XML format facilitates, in this case, presenting the results of the program on the Web, and allows quicker access to information. XML (the Extensible Markup Language) was first ratified by the W3C (World Wide Web Consortium) as the standard for information exchange on the Internet in February 1998. XML specifies a rigorous, text-based way to represent the structure inherent in data so that it can be authored and interpreted unambiguously.

What is the default charset used by the program (when parameter CS is not specified in the command line)?

The default charset (when parameter CS is not specified in the command line) is iso-8859-1.

How do you perform (what Java statements are used) I/O operations on files and how do you read/write data in various languages?

The I/O operations on files which enable working with data in various languages (coded with various character sets) have been written by using the Java specifications `InputStreamReader`, `BufferedReader`, and `OutputStreamWriter`, `BufferedWriter` respectively.

What functions for handling strings are used in the program?

The program uses both Java functions for handling strings and user-defined methods for parsing rows of text read from the dictionaries, and for extracting strings that are significant for the algorithm (for instance, word translations).

What Java classes are defined in the program and what do they contain?

The program defines a unique class, `GenSynsets`, which contains all variables and methods necessary in the implementation of the algorithms on which it is based. Of the defined methods we would like to mention those referring to I/O operations using the dictionary files, those for the labeling of the generated e-sets, the method which implements the backtracking-type strategy, sorting operations and the operations on strings.

How does the program determine the gloss corresponding to an English synset?

The program finds the gloss corresponding to an English synset by using the `getGloss()` method which belongs to the `Synset` class.

How does the program determine the final synsets corresponding to the chosen foreign language? (asked twice)

In order to determine the final synsets (corresponding to the chosen foreign language) the program combines the e-sets labeled with the maximum value, corresponding to each eword, and eliminates all duplicate words. This combining is implemented by means of the backtracking method.

How does the program deal with the situation in which there are words of a synset, specified by means of its offset, for which no corresponding entries exist in the English-Foreign (Language) Dictionary? (asked twice)

In this situation only those words of the synsets are used for which a translation exists (namely for which corresponding entries in the English-Foreign Language Dictionary exist).

What programming techniques does your program make use of? (asked twice)

Among the programming techniques which we have used the most important is the Backtracking method by means of which the fsynsets are generated, starting from the esets, and with elimination of duplicates.

Have you used any of the results of previous WordNet projects implemented in Java? (asked twice)

Our program GenSynsets uses results which were obtained within the project JWordNet. JwordNet is a pure Java standalone object-oriented interface to the WordNet database of lexical relationships. It is intended for Java programmers who wish to write portable Java applications that use a local copy of the WordNet files, or who find JWordNet's object-oriented interface preferable to the procedural interface that the C library (and native method interfaces built on top of it) provide. It includes the Dictionary broker class, the IndexWord, Synset, Word, and Pointer domain entity classes, and the POS and PointerType enumeration holders.

How does the program go over all the synsets of the American WordNet? (asked twice)

In this case, going over all the synsets of the American WordNet is based on an enumeration and on the method synsets which belongs to the class DictionaryDatabase (JWordNet).

How does the program determine the words of a synset specified by means of the corresponding offset? (asked twice)

When running the program according to a list of synsets (specified by the corresponding offsets) two arguments are passed to the method: POS (part of speech) and the offset of a synset. Finding the synsets of the American WordNet

relies on the method `getSynsetAt()` which belongs to the class `FileBackedDictionary` (`JWordNet`).

Questions Concerning the RORIC-LING Implementation of WN Algorithms

Why are there no numerals following certain items in your own coding of adjective clusters? (asked twice)

Because, for the sake of simplicity, within this demo we have worked with cleared, combined synsets where we have given up certain parameters like the sense number. The existing algorithm can be easily modified in order for it to work with synsets where this parameter exists and to take it into account. Also, in order to generate completely coded WN-type adjective clusters in a foreign language, one must first obtain all adjective synsets of that language, in order to see all existing meanings of a certain adjective in the target language. This is something we haven't done yet for Romanian, due to the existing incomplete bilingual dictionaries in electronic format. On the other hand, our claim was not to have generated clusters identical to those in WN, but just WN-type adjective synsets and clusters. Again, the existing computer programs can be easily modified in order to obtain the exact form of a WN adjective cluster.

As it is well known, many adjectives are limited as to the syntactic positions they can occupy. Is this limitation coded in WordNet and how? Have you performed this coding in your own implementation? (asked twice)

As already mentioned, many adjectives are limited as to the syntactic positions they can occupy, and that limitation is usually coded in WordNet. Because it is a word-form limitation, it is coded for individual adjectives rather than for synsets. Consider, for instance, the cluster **awake/asleep**, both of which are limited to predicate position. Although these are the head words of the cluster, the limitation does not hold for all of the synonyms in the cluster. Therefore, the individual words so limited are all coded with (p). For adjectives limited to prenominal (attributive) position, the code is (a). And, finally, for those few adjectives that can appear only immediately following a noun, the code is (ip) for "immediately postnominal". This codification has not yet been implemented for Romanian adjective clusters.

Why don't you discuss verbs in WN as well, within the framework of this project?

Because this is not an exhaustive discussion concerning WordNet and/or the semiautomatic generation of WordNets for other languages. We have chosen to discuss the two basic WordNet structures - namely the hierarchy and the cluster - for which we have studied nouns and adjectives in WordNet. For a detailed discussion concerning all WordNet translation issues see the project BALKANet at

<http://www.ceid.upatras.gr/Balkanet>

Can you mention some of the difficulties you encountered when implementing the described algorithm for your own language - Romanian? (asked twice)

The main difficulties which occurred when automatically translating the English synsets into Romanian ones were generated by the so-called "false friends", by collocations, by loan translation, and by the fact that the polysemy of many English words is greatly superior to that of the corresponding Romanian words. For detailed explanations concerning all these phenomena please read the linguistic comments that you will find in the web page of the project.

General Questions Concerning WordNet

What is the so-called "lexical matrix" in WordNet?

The lexical matrix is a matrix in which word forms are imagined to be listed as headings for the columns; word meanings are headings for the rows. An entry in a cell of the matrix implies that the form in that column can be used (in an appropriate context) to express the meaning in that row. If there are two entries in the same column, the word form is polysemous; if there are two entries in the same row, the two word forms are synonyms (relative to a context). For more information see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look under "Publications" at the address <http://www.cogsci.princeton.edu/~wn/>

Does WordNet distinguish between semantic relations and lexical relations?

WordNet distinguishes between semantic relations and lexical relations; the emphasis is still on semantic relations between meanings, but relations between

words are also included. However, WordNet is organized by semantic relations, which are indicated by pointers.

What do you think made it necessary to partition WN into nouns, verbs, adjectives and adverbs?

Synonymy is the central relation in WN. The definition of synonymy in terms of substitutability makes it necessary to partition WordNet into nouns, verbs, adjectives and adverbs.

As it is noted in "Five Papers on WordNet", "if concepts are represented by synsets, and if synonyms must be interchangeable, then words in different syntactic categories cannot be synonyms (cannot form synsets) because they are not interchangeable. Nouns express nominal concepts, verbs express verbal concepts, and modifiers provide ways to qualify those concepts. In other words, the use of synsets to represent word meanings is consistent with psycholinguistic evidence that nouns, verbs, and modifiers are organized independently in semantic memory".

Synonymy and antonymy are lexical relations between word forms. How about hyponymy and hypernymy?

Unlike synonymy and antonymy, which are lexical relations between word forms, hyponymy/hypernymy (also called subordination/superordination, subset/superset, or the ISA relation) is a semantic relation between word meanings.

Hyponymy is transitive and asymmetrical (Lyons, 1977, vol. 1), and, since there is normally a single superordinate, it generates a hierarchical semantic structure, in which a hyponym is said to be below its superordinate. Such hierarchical representations are widely used in the construction of information retrieval systems, where they are called inheritance systems (Touretzky, 1986): a hyponym inherits all the features of the more generic concept and adds at least one feature that distinguishes it from its superordinate and from any other hyponyms of that superordinate. This convention provides the central organizing principle for the nouns in WordNet.

Please explain briefly the meronymy relation and indicate where it is present in WordNet.

A semantic relation in WordNet is the part-whole (or HASA) relation, known to lexical semanticists as meronymy/holonymy. The meronymic relation is transitive (with qualifications) and asymmetrical (Cruse, 1986), and can be used to construct a part hierarchy (with some reservations, since a meronym can have many holonyms). It is assumed that the concept of a part of a whole can be a part of a

concept of the whole, although it is recognized that the implications of this assumption deserve more discussion than they receive within this framework. Meronymy is present in the organization of noun synsets.

Are there any morphological relations in WordNet?

As it is noted in "Five Papers on WordNet", "an important class of lexical relations are the morphological relations between word forms. Initially, interest was limited to semantic relations; no plans were made to include morphological relations in WordNet. As work progressed, however, it became increasingly obvious that if WordNet was to be of any practical use to anyone, it would have to deal with inflectional morphology. For example, if someone put the computer's cursor on the word **trees** and clicked a request for information, WordNet should not reply that the word was not in the database. A program was needed to strip off the plural suffix and then to look up **tree**, which certainly is in the database. This need led to the development of a program for dealing with inflectional morphology".

What is the main difference between a prototypical definition recorded in paper dictionaries and a WN definition of nouns, for instance?

The prototypical definition points upward, to a superordinate term, not sideways to coordinate terms or downward to hyponyms. For instance, the definition of **tree** in standard paper dictionaries points to the superordinate term **plant**, but contains no information about coordinate terms. A dictionary definition draws some important distinctions and serves to remind the reader of something that is presumed to be familiar already; it is not intended as a catalogue of general knowledge.

What can you tell me about the semantic relation that has been called the ISA relation and about its implementation relatively to nouns in WN?

The semantic relation that is represented in WordNet by '@->' has been called the ISA relation, or the hypernymic or superordinate relation (since it points to a hypernym or superordinate term); it goes from specific to generic and so is a generalization. The inverse semantic relation '~-->' goes from generic to specific (from superordinate to hyponym) and so is a specialization.

As it is noted in "Five Papers on WordNet", "since a noun usually has a single superordinate, dictionaries include the superordinate in the definition; since a noun can have many hyponyms, English dictionaries do not list them (the French dictionary "Le Grand Robert" is an exception). Even though the specialization relation is not made explicit in standard dictionaries of English, it is a logical derivative of the generalization relation. In WordNet, lexicographers code the generalization relation '@->' explicitly with a labeled pointer between lexical

concepts or senses. When the lexicographers' files are converted automatically into the lexical database, one step in this process is to insert inverse pointers for the specialization relation '~>'. Thus, the lexical database is a hierarchy that can be searched upward or downward with equal speed." Computer scientists call such hierarchies "inheritance systems", because they think of specific items inheriting information from their generic superordinates. All the properties of the superordinate are assumed to be properties of the subordinate as well; instead of listing those properties redundantly with both items, they are listed only with the superordinate and a pointer from the subordinate to the superordinate is understood to mean "for additional properties, look here".

It is said that WN is a lexical inheritance system. Please give an example in the case of nouns and explain the corresponding WN implementation.

WN is indeed a lexical inheritance system. A systematic effort has been made to connect hyponyms with their superordinates (and vice versa). In the WN data base, an entry for **tree**, for instance, contains a reference, or pointer '@->', to an entry for **plant**; the pointer is labeled "superordinate" by the arbitrary symbol '@'. In the database, the pointer '@' to the superordinate **plant** will be reflected by an inverse pointer '~' to **tree** in the synset for **plant**; that pointer is labeled "hyponym" by the arbitrary symbol '~'. The computer is programmed to use these labeled pointers to construct whatever information a user requests; the arbitrary symbols '@' and '~' are suppressed when the requested information is displayed. The synset for tree would look something like:

{tree,plant,@ conifer,~alder,~... }

where the '~...' is filled with many more pointers to hyponyms. The synset for plant would look something like

{plant,flora,organism,@ tree,~... }.

Is there psycholinguistic evidence that people's lexical memory for nouns forms an inheritance system?

The first person to make this claim explicit seems to have been Quillian (1967, 1968). Experimental tests of Quillian's proposal were reported in a seminar paper by Collins and Quillian (1969), who assumed that reaction times can be used to indicate the number of hierarchical levels separating two meanings.

An alternative conclusion - the conclusion on which WordNet is based - is that the inheritance assumption is correct, but that reaction times do not measure what

Collins and Quillian, as well as other experimentalists, assumed they did. Perhaps reaction times indicate a pragmatic rather than a semantic distance - a difference in word use, rather than a difference in word meaning (Miller and Charles, 1991).

Are all nouns contained in a single hierarchy in WN? (asked twice)

In WN the nouns are partitioned with a set of semantic primes, namely a relatively small set of generic concepts is selected and each one of them is treated as the unique beginner of a separate hierarchy. These multiple hierarchies correspond to relatively distinct semantic fields, each with its own vocabulary.

WN has adopted the following set of 25 unique beginners:

- | | |
|-------------------------|-----------------------|
| {act, action, activity} | {natural object} |
| {animal, fauna} | {natural phenomenon} |
| {artifact} | {person, human being} |
| {attribute, property} | {plant, flora} |
| {body, corpus} | {possession} |
| {cognition, knowledge} | {process} |
| {communication} | {quantity, amount} |
| {event, happening} | {relation} |
| {feeling, emotion} | {shape} |
| {food} | {state, condition} |
| {group, collection} | {substance} |
| {location, place} | {time} |
| {motive} | |

The most important criterion in choosing these primitive semantic components is that, collectively, they should provide a place for every English noun. The resulting hierarchies vary widely in size and are not mutually exclusive - some cross-reference is required - but on the whole they cover distinct conceptual and lexical domains. They were selected after considering the possible adjective-noun combinations that could be expected to occur (an analysis carried out by Philip N. Johnson-Laird).

What do "generic concepts" mean with respect to nouns in WordNet?

The hierarchies of nominal concepts in WN are said to have a level, somewhere in the middle, where most of the distinguishing features are attached. It is referred to as the basic level, and the nominal concepts at this level are called basic-level categories or generic concepts (Berlin, Breedlove, and Raven, 1966, 1973). Rosch (1975; Rosch, Mervis, Gray, Johnson, and Boyes-Braem, 1976) extended this generalization: for concepts at the basic level, people can list many distinguishing

features. Above the basic level, descriptions are brief and general. Below the base level, little is added to the features that distinguish basic concepts.

Do you think it is possible to identify alternative senses of a word only by the use of synonyms? How does WN cope with this problem? (asked twice)

As it is noted in "Five Papers on WordNet", "as the coverage of WordNet increased, it became increasingly obvious that alternative senses of a word could not always be identified by the use of synonyms. Rather late in the game, therefore, it was decided to include distinguishing features in the same way that conventional dictionaries do, by including short explanatory glosses as a part of synsets containing polysemous words. These are marked off from the rest of the synset by parentheses".

Are meronyms distinguishing features that hyponyms inherit in WN?

As it is noted in "Five Papers on WordNet", "meronyms are distinguishing features that hyponyms can inherit. Consequently, meronymy and hyponymy become intertwined in complex ways. For example, if **beak** and **wing** are meronyms of **bird**, and if **canary** is a hyponym of **bird**, then, by inheritance, **beak** and **wing** must also be meronyms of **canary**".

Can parts be hyponyms as well as meronyms? If yes, please give an example from WN.

The connections between meronymy and hyponymy are complicated by the fact that parts are hyponyms as well as meronyms. The example which is given in "Five Papers on WordNet" is the synset {beak, bill, neb}, which is a hyponym of {mouth, muzzle}, which in turn is a meronym of {face, countenance} and a hyponym of {orifice, opening}. A frequent problem in establishing the proper relation between hyponymy and meronymy arises from a general tendency to attach features too high in the hierarchy. For example, if wheel is said to be a meronym of vehicle, then sleds will inherit wheels they should not have. Indeed, in WN a special synset was created for the concept {wheeled vehicle}.

In what hierarchies of WordNet is meronymy mostly found?

Meronyms tend to occur most frequently in connection with words denoting physical objects. In WN, meronymy is found primarily in the {body, corpus}, {artifact}, and {quantity, amount} hierarchies.

Is it true that the "part-of" relation is transitive?

The "part-of" relation is often compared to the "kind of" relation: both are asymmetric and (with reservations) transitive, and can relate terms hierarchically (Miller and Johnson-Laird, 1976). That is to say, parts can have parts: a finger is a part of a hand, a hand is a part of an arm, an arm is a part of a body: the term **finger** is a meronym of the term **hand**, **hand** is a meronym of **arm**, **arm** is a meronym of **body**. But the "part of" construction is not always a reliable test of meronymy. In many instances transitivity seems to be limited (Lyons, 1977).

For more information see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

Is it true that there are various types of "part-of" relations in general? What is the situation of their implementation in WN?

Winston et al.(1987) differentiate six types of meronyms: component-object (branch/tree), member-collection (tree/forest), portion-mass (slice/cake), stuff-object (aluminium/airplane), feature-activity (paying/shopping), and place-area (Princeton/New Jersey). Chaffin, Hermann, and Winston (1988) add a seventh: phase-process (adolescence/growing up). Meronymy is obviously a complex semantic relation - or set of relations. Only three of these types of meronymy are coded in WN: "is a component part of", "is a member of", and "is the stuff that it is made from". Of these three, the "is a component of" relation is by far the most frequent.

Does the antonymy relation exist between nouns and, if so, how is it represented in WN? (asked twice)

As it is noted in "Five Papers on WordNet", "semantic opposition is not a fundamental organizing relation between nouns, but it does exist and so merits its own representation in WordNet. For example, the synsets for **man** and **woman** would contain:

{ [man, woman,!], person,@ ... (a male person) }

{ [woman, man,!], person,@ ... (a female person) }

where the symmetric relation of antonymy is represented by the '!' pointer, and

square brackets indicate that antonymy is a lexical relation between words, rather than a semantic relation between concepts.

What are the main semantic relations which are taken into consideration in WN with respect to nouns?

The main semantic relations which are taken into consideration in WN with respect to nouns are hyponymy, meronymy, and antonymy. When all these three kinds of semantic relations are included, the result is a highly interconnected network of nouns.

Do the adjective synsets in WN contain only adjectives?

The adjective synsets in WN contain mostly adjectives, although some nouns and prepositional phrases that function frequently as modifiers have been entered as well. The discussion within RORIC-LING is limited to adjectives.

What are the major classes of adjectives taken into consideration in WordNet?

WN divides adjectives into two major classes: descriptive and relational. Descriptive adjectives ascribe to their head nouns values of (typically) bipolar attributes and consequently are organized in terms of binary oppositions (antonymy) and similarity of meaning (synonymy). Descriptive adjectives that do not have direct antonyms are said to have indirect antonyms by virtue of their semantic similarity to adjectives that do have direct antonyms. WN contains pointers between descriptive adjectives expressing a value of an attribute and the noun by which that attribute is lexicalized. Reference-modifying adjectives have special syntactic properties that distinguish them from other descriptive adjectives. Relational adjectives are assumed to be stylistic variants of modifying nouns and so are cross-referenced to the noun files. Chromatic color adjectives are regarded as a special case.

What exactly is a descriptive adjective?

A descriptive adjective is one that ascribes a value of an attribute to a noun. That is to say, **x is Adj** presupposes that there is an attribute A such that **A(x)=Adj**. To say "The package is heavy" presupposes that there is an attribute WEIGHT such that **WEIGHT (package) = heavy**. Similarly, **low** and **high** are values for the attribute HEIGHT. WordNet contains pointers between descriptive adjectives and the noun synsets that refer to the appropriate attributes.

Ws the semantic organization of descriptive adjectives in WN similar in any way to that of nouns? (asked twice)

The semantic organization of descriptive adjectives is entirely different from that of nouns. There is no relation generating nominal hierarchies in the case of adjectives. The semantic organization of adjectives is more naturally thought of as an abstract hyperspace of N dimensions rather than as a hierarchical tree.

Is the basic semantic relation among adjectives in WN the antonymy relation or the similarity relation and how is it represented in WN? (asked twice)

The basic semantic relation among descriptive adjectives is antonymy. The importance of antonymy first became obvious from results obtained with word association tests. The importance of antonymy in the organization of descriptive adjectives is understandable when it is recognized that the function of these adjectives is to express values of attributes, and that nearly all attributes are bipolar. Antonymous adjectives express opposing values of an attribute. For example, the antonym of **heavy** is **light**, which expresses a value at the opposite pole of the WEIGHT attribute. In WordNet, this binary opposition is represented by reciprocal labeled pointers: **heavy!->light** and **light!->heavy**.

Can the antonymy relation be so important considering that many descriptive adjectives have no antonyms? (asked twice)

Because many descriptive adjectives have no antonyms WN has introduced a similarity pointer and has used it to indicate that the adjectives lacking antonyms are similar in meaning to adjectives that do have antonyms. Gross, Fischer, and Miller (1989) proposed that adjective synsets be regarded as clusters of adjectives associated by semantic similarity to a focal adjective that relates the cluster to a contrasting cluster at the opposite pole of the attribute. Gross, Fischer and Miller distinguish direct antonyms like **heavy/light**, which are conceptual opposites that are also lexical pairs, from indirect antonyms, like **heavy/weightless**, which are conceptual opposites that are not lexically paired. Under this formulation, all descriptive adjectives have antonyms; those lacking direct antonyms have indirect antonyms, i.e., are synonyms of adjectives that have direct antonyms.

Does the adjective organization of WN represent a claim that all descriptive adjectives have antonyms? (asked twice)

Some descriptive adjectives do not have direct antonyms. However, in the adjective organization of WN, those lacking direct antonyms have indirect antonyms, i.e., are synonyms of adjectives that have direct antonyms. Under this formulation, all descriptive adjectives have antonyms.

How are indirect antonyms established in WordNet?

In WN those adjectives lacking direct antonyms have indirect antonyms, i.e., are synonyms of adjectives that have direct antonyms. Direct antonyms are represented by an antonymy pointer, '!->'; indirect antonyms are inherited through similarity, which is indicated by the similarity pointer, '&->'.

What is, in short, the basic model presented by the authors of WN with respect to adjectives?

The basic model presented by the authors of WN with respect to adjectives consists of dividing adjectives into two major types, descriptive (which enter into clusters based on antonymy) and relational (which are similar to nouns used as modifiers). Without claiming complete coverage, the authors of WN believe that this model accounts for the majority of English adjectives.

What do you know about the conceptually important relation of gradation and has it been coded in WN?

A gradable adjective can be defined as one whose value can be multiplied by such adverbs of degree as very, decidedly, intensely, rather, quite, somewhat, pretty, extremely (Cliff, 1959).

Gradation must also be considered as a semantic relation organizing lexical memory for adjectives (Bierwisch, 1989). For some attributes gradation can be expressed by ordered strings of adjectives, all of which point to the same attribute noun in WordNet.

As it is noted in "Five Papers on WordNet", "it would not be difficult to represent ordered relations by labeled pointers between synsets, but it was estimated that not more than 2% of the more than 2,500 adjective clusters could be organized in that

way. Since the conceptually important relation of gradation does not play a central role in the organization of adjectives, it has not been coded in WordNet."

Is there any connection in WN between the noun expressing an attribute and the adjectives expressing values of that attribute? (asked twice)

The noun that names the attribute - e.g., LENGTH - and all the adjectives expressing values of that attribute (in this case long, short, lengthy, etc.) are linked in WordNet by a pointer.

How are the names of colors introduced in WordNet?

In WN the opposition **colored/colorless** (cross-referenced to **chromatic/achromatic**) is used to introduce the names of colors. Hues are coded as similar to colored, and the shades of gray from white to black are coded as similar to **gray**, which is in a tripartite cluster with white and black, providing for a graded continuum.

What are relational adjectives?

Relational adjectives, which were first discussed at length by Levi (1978), mean something like "of, relating/pertaining to, or associated with" some noun, and they play a role similar to that of a modifying noun.

For example, **fraternal** as in **fraternal twins** relates to **brother**, and **dental** as in **dental hygiene**, is related to **tooth**.

What are the main differences between relational adjectives and descriptive adjectives? (asked twice)

The main differences are the following:

1. Relational adjectives differ from descriptive adjectives in that they do not relate to an attribute.
2. Relational adjectives do not refer to a property of their head nouns.
3. Relational adjectives, like nouns and unlike descriptive adjectives, are not gradable.
4. Relational adjectives do not have direct antonyms; although they can often be combined with non-, such forms do not express the opposite value of an attribute but something like "everything else". Since relational adjectives do not have antonyms, they cannot be incorporated into the clusters that characterize descriptive adjectives.

WordNet maintains a separate file of relational adjectives with pointers to the corresponding nouns. For more details see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

How does WordNet treat relational adjectives? (asked twice)

WordNet maintains a separate file of relational adjectives with pointers to the corresponding nouns.

Some 1,700 relational adjective synsets containing over 3,000 individual lexemes are currently included in WordNet. Each synset consists of one or more relational adjectives, followed by a pointer to the appropriate noun.

For more details see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

What do the numerals following certain items in the coding of WN adjective clusters stand for? (asked twice)

The numerals following certain items distinguish different subsenses or different privileges of occurrence - for example, the dried-up₁ of a water hole in one synset and the dried-up₂ of autumn leaves or fruit in another. Each of these cases, furthermore, contains parenthetical information designed to help distinguish these particular senses or indicate acceptable contexts.

Do adjective clusters contain pointers to other related clusters?

As it is noted in "Five Papers on WordNet", "in addition to the lowercase within-cluster pointers, many head synsets contain pointers to other, related clusters. In the AWAKE/ASLEEP cluster, the capitalized pointer ALERT,& points to the head word of the ALERT/UNALERT cluster. These capitalized pointers are planned to serve as "see also" cross-references to related clusters.

What can you tell me about adjective clusters headed by two pairs of adjectives in WordNet? (asked twice)

The restricted within-cluster coding leads to a problem when closely related attributes are expressed by more than one pair of antonyms. In such cases, exactly the same set of synsets can be related to two different antonymous pairs, some of

which are presently in different clusters. (Consider large/small and big/little). In such cases a single cluster has been created headed by both pairs, thus avoiding unnecessary redundancy. In addition, a particular synset can be coded with two pointers, one to its own cluster head, the other to the head of an outside cluster.

Are verbs organized in WordNet according to what linguists call "semantic domains"?

As it is explained in "Five Papers on WordNet", "verbs are divided into 15 files, largely on the basis of semantic criteria. All but one of these files correspond to what linguists have called semantic domains: verbs of bodily care and functions, change, cognition, communication, competition, consumption, contact, creation, emotion, motion, perception, possession, social interaction, and weather verbs. Virtually all the verbs in these files denote events or actions. Another file contains verbs referring to states, such as **suffice**, **belong**, and **resemble**, that could not be integrated into the other files. The verbs in this latter group do not constitute a semantic domain, and share no semantic properties other than that they refer to states. This file, whose organization resembles that of the adjectives in WordNet, consists of small semantic clusters. The division of verbs into 14 files corresponding to different semantic domains, each containing event and action verbs, and one file containing semantically diverse stative verbs reflects the division between the major conceptual categories EVENT and STATE found in Jackendoff's (1983) and Dowty's (1979) analyses".

What main principles underlie the semantic relations between nouns, adjectives and verbs in WN? (asked twice)

The principle of lexical inheritance can be said to underlie the semantic relations between nouns, and bipolar oppositions serve to organize the adjectives. Similarly, the different relations that organize the verbs can be cast in terms of one overarching principle, lexical entailment (or strict implication).

How does the entailment relation between verbs in WN resemble meronymy between nouns? (asked twice)

The entailment relation between verbs resembles meronymy between nouns, but meronymy is better suited to nouns than to verbs. The following example concerning verbs is offered in "Five Papers on WordNet":

"Snoring or dreaming can be a part of sleeping, in the sense that the two activities are, at least partially, temporally co-extensive: the time that you spend snoring or dreaming is a proper part of the time you spend sleeping. And it is true that when you stop sleeping you also necessarily stop snoring or dreaming."

A verb X will be said to include a verb Y if there is some stretch of time during which the activities denoted by the two verbs co-occur, but no time during which Y occurs and X does not. If there is a time during which X occurs but Y does not, X will be said to properly include Y. A simple generalization of this would be the following: if X entails Y, and if a temporal inclusion relation holds between them, then people will accept a part-whole statement relating Y and X.

What can you tell me about hyponymy among verbs in WN? (asked twice)

The sentence frame used to test hyponymy between nouns, **An x is a y**, is not suitable for verbs, because it requires that **x** and **y** be nouns. The semantic distinction between two verbs is different from the features that distinguish two nouns in a hyponymic relation.

The many different kinds of elaborations that distinguish a 'verb hyponym' from its superordinate have been merged into a manner relation that Fellbaum and Miller (1990) have dubbed troponymy (from the Greek tropos, manner or fashion). The troponymy relation between two verbs can be expressed by the formula **To X is to Y in some particular manner**.

For more details see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

Is troponymy a special case of entailment (with regard to verbs in WordNet)? (asked twice)

Troponymy is a particular kind of entailment, in that every troponym X of a more general verb Y also entails Y. Consider the pair **limp-walk**, which represents the example offered in "Five Papers on WordNet". The authors comment this example as follows: "The verbs in this pair are related by troponymy: **to limp** is also **to walk in a certain manner**; **limp** is a troponym of **walk**. The verbs are also in an entailment relation: the statement **He is limping** entails **He is walking**, and walking can be said to be a part of limping. Unlike the activities denoted by **snore** and **sleep**, or **buy** and **pay**, the activities referred to by a troponym and its more general superordinate are always temporally co-extensive, in that one must necessarily be walking every instant that one is limping. Troponymy therefore represents a special case of entailment: pairs that are always temporally co-extensive and are related by entailment".

Is it true that there are various kinds of entailment with temporal inclusion in WordNet?

Two kinds of entailment with temporal inclusion are accounted for in WN. One type of entailment is troponymy (limp-walk), while entailment without troponymy refers to pairs of verbs (snore-sleep) related only by entailment and proper temporal inclusion.

What can you tell me about opposition and entailment with regard to the semantic organization of verbs in WordNet?

As it is noted in "Five Papers on WordNet", "many verb pairs in an opposition relation also share an entailed verb. For example, both **hit** and **miss** entail **aim**, because one must necessarily aim in order to hit or miss". In contrast to other kinds of entailment, "these verbs are not related by temporal inclusion. The activities denoted by **hit** (or **miss**) and **aim** occur in a sequential order: in order to either hit or miss, one must have aimed first; aiming is a precondition for both hitting and missing." For more information see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

How many types of entailment relations among verbs have been taken into consideration in WordNet? (asked twice)

The four types of entailment relations among verbs that have been taken into consideration in WordNet are the following:

- entailment + temporal inclusion + troponymy (limp-walk);
- entailment + temporal inclusion - troponymy (snore-sleep, buy-pay);
- entailment - temporal inclusion + backward presupposition (succeed-try, untie-tie);
- entailment - temporal inclusion + cause (raise-rise, give-have).

For more information see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

Are some or all syntactic aspects of verbs covered in WordNet and how?

As it is explained in "Five Papers on WordNet", to cover at least the most important syntactic aspects of verbs "WordNet includes for each verb synset one or several sentence frames, which specify the subcategorization features of the verbs in the synset by indicating the kinds of sentences they can occur in. This information permits one to quickly search among the verbs for the kinds of semantic-syntactic regularities studied by Levin and others."

For more information see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

Which are some specific verb files and clusters specified in the American WordNet?

Here are the main verb files in the American WordNet:

- Verbs of bodily functions and care (sweat, shiver, faint, ache, tire, sleep, freeze);
- Verbs of change (change, alter, vary, modify);
- Verbs of communication (beg, order, thank);
- Competition verbs (fight, race);
- Consumption verbs (drink, eat);
- Contact verbs (scrub, wipe, squeeze);
- Cognition verbs (reasoning, judging, learning, memorizing);
- Creation verbs (engrave, sew, bake);
- Motion verbs (move, travel);
- Emotion verbs (amuse, encourage);
- Stative verbs (surround, cross, reach);
- Perception verbs (watch, spy, survey);
- Verbs of possession (hold, own, give, transfer, take, receive);
- Verbs of social interaction (impeach, excommunicate);
- Weather verbs (rain, thunder, snow).

For more information see "Five Papers on WordNet" which is available in PostScript, Acrobat (PDF), or as a compressed tar file containing the nroff source. Please look at <http://www.cogsci.princeton.edu/~wn/> under "Publications".

III

A THEORETICAL SPECIFICATION CONCERNING A MORPHOLOGICAL MODEL FOR ROMANIAN

1. A few remarks on Romanian morphology

Many comments could be made regarding Romanian morphology, as it is much more intricate than English morphology and differs from the latter in point of origin. One should not forget that English is a Germanic language, while Romanian comes from Latin, and Latin in its turn had a very intricate morphology. Another specification that is required is the fact that English morphology is far more developed and due to this fact is simpler and more systematic. Romanian morphology, for instance, has a rich and varied verbal inflection. In this respect it suffices to specify that the great Swedish Romanist Alf Lombard has dedicated a vast monograph in two volumes to the Romanian verb (over 1200 pages). In “Grammar of the Romanian Language”, published by the Romanian Academy and whose point of view underlies the present paper, it is admitted that there are four verbal conjugations in Romanian. However, there are Romanian and foreign researchers who consider that we have 6, 8, 10, 12 and even 14 conjugations.

Romanian displays very many different pronominal forms and adjectival classes (the latter having 4 suffixes, 3 suffixes, only 2 suffixes, as well as a few categories of non-flexible adjectives). According to certain researchers there are about 10 declensions for the Romanian adjective. We also have four types of articles out of which the most important is the definite article. It is usually post-posed (i.e. located at the end of the word), but there are many nouns that can receive, in Genitive – Dative, a pre-posed article.

The Romanian noun is also extremely intricate, because Romanian exhibits three genders: masculine, feminine and neuter. Romanian is the only language of Latin origin that has retained the neuter gender, most likely under the influence of Slavic. Having three genders, Romanian also has a rich and varied nominal

inflection. It is especially the plural of Romanian nouns that is formed with various endings, to which various differentiating markers are added, among them the phonetic alternations (vocalic and consonantic) being the most important. An example (among many others) is given by the feminine noun stradă ('street') that has the plural form străzi. Another example is the noun roată ('wheel') that has the plural form roți. As one can notice, the plural differs from the singular by the i ending (opposed to the singular ă), by the vocalic alternation a/ă and the consonantic one d/z. In the second case the vocalic alternation is oa/o, and the consonantic one is t/ț, triggered (as it was the case in the first example) by the presence of i. There are hundreds of Romanian nouns that have two plural forms, but, in most cases, only one is accepted by the norms of literary language. In other cases both forms are correct, something that does not happen in English, where the plural form of nouns is usually regular and is formed by the addition of an s to the singular form.

In what follows we shall proceed according to the point of view of the "Grammar of the Romanian Language" published by the Romanian Academy and we shall exemplify the inflectional approach to morphology in the case of Romanian nouns and adjectives.

2. The paradigm

As it is noted in [3], "the non-flexible words have a unique form of representation and are easy to analyze from a lexical point of view. The flexible ones change their form in different syntactic situations, i.e. they can be declined or conjugated. The sum of all the flexionary forms of a word represents its paradigm." In what follows, the paradigm that we shall take into consideration is the following:

P:: {Stem + Flective}

Within this paradigm the stem may vary due to phonetic alternations or irregular forms. We would like to emphasize the fact that we are using

the term *stem* and not that of *root*. By ‘stem’, ‘theme’ or ‘thema’ linguists denote the stem of the word to which affixes / infixes can be added. Thus, the stem represents the inflexional base of a word to which other elements, such as thematic vowels and consonants, inflections etc. are added. Obviously, in many cases, the stem of a word can be identical to its root. More details concerning the use of this terminology in Romanian can be found in [2].

Unlike the stem, which is compulsory, the flective can be zero. Again according to [3], the flective varies according to the grammatical categories of the part of speech. In the case of nouns, for instance, this variation is determined by number, case, gender and determination. In the case of verbs it corresponds to mood, tense, number and person. Within the same part of speech, the flective corresponding to different grammatical categories can have different forms. This is the criterion according to which words that are different but which belong to the same part of speech are grouped in flexionary classes. The flectives that characterize a specific flexionary class are the same for all words belonging to that class.

When studying the generation of Romanian noun and adjective forms it has been considered (see 3.) that the flective either is zero, either is formed of a connecting vowel and a definite article (whenever it may be the case) or of an inflectional ending and a definite article (whenever it may be the case).

3. Generation of Romanian noun and adjective forms

3.1. NOUNS

In the case of Romanian nouns three genders exist (masculine, feminine and neuter) and two main types of inflection are known: inflection with a definite article and inflection with a non- definite article. In the case of each following rule inflexion with a definite article as well as with a non-definite article are shown. The following abbreviations have been used: 'N' for nominative, 'AC' for accusative, 'G' for genitive, 'D' for dative, 'sg.' for singular, 'pl.' for plural, 'Infl.' for **inflectional ending**, 'Conv.' for **connecting vowel** and 'Art.' for article.

1.a. MASCULINE NOUNS

Rule no. 1

Definite article

sg.

N + AC: Stem + Conv. - u + Art. - l

G + D: Stem + Conv. - u + Art. - lui

pl.

N + AC: Stem + Infl. - i + Art. - i

G + D: Stem + Infl. - i + Art. - lor

Non - definite article

sg.

N + AC: Stem

G + D: Stem

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

Rule no. 2

Definite Article

sg.

N + AC: Stem + Infl. - **e** + Art. - **le**

G + D: Stem + Infl. - **e** + Art. - **lui**

pl.

N + AC: Stem + Infl. - **i** + Art. - **i**

G + D: Stem + Infl. - **i** + Art. - **lor**

Non - definite article

sg.

N + AC: Stem + Infl. - **e**

G + D: Stem + Infl. - **e**

pl.

N + AC: Stem + Infl. - **i**

G + D: Stem + Infl. - **i**

1.b. NEUTER NOUNS

Rule no.3

Definite article

sg.

N + AC: Stem + Conv. - **u** + Art. - **l**

G + D: Stem + Conv. - **u** + Art. - **lui**

pl.

N + AC: Stem + Infl. - **uri** + Art. - **le**

G + D: Stem + Infl. - **uri** + Art. - **lor**

Non - definite article

sg.

N + AC: Stem

G + D: Stem

pl.

N + AC: Stem + Infl. - **uri**

G + D: Stem + Infl. - **uri**

Rule no. 4

Definite article

sg.

N + AC: Stem + Conv. - **u** + Art. - **l**

G + D: Stem + Conv. - **u** + Art. - **lui**

pl.

N + AC: Stem + Infl. - **e** + Art. - **le**

G + D: Stem + Infl. - **e** + Art. - **lor**

Non - definite article

sg.

N + AC: Stem

G + D: Stem

pl.

N + AC: Stem + Infl. - **e**

G + D: Stem + Infl. - **e**

EXCEPTION (Rule specifics):

- change of vowel in the stem : N,G,D,AC pl. - **o/oa**

Rule no. 5

Definite article

sg.

N + AC: Stem + Conv. - **u** + Art. - **l**

G + D: Stem + Conv. - **u** + Art. - **lui**

pl.

N + AC: Stem + Infl. - **i** + Art. - **le**

G + D: Stem + Infl. - **i** + Art. - **lor**

Non - definite article

sg.

N + AC: Stem + Infl. - **u**

G + D: Stem + Infl. - **u**

pl.

N + AC: Stem + Infl. - **i**

G + D: Stem + Infl. - **i**

Rule no. 6

Definite article

sg.

N + AC: Stem + Infl. - **e** + Art. - **le**
G + D: Stem + Infl. - **e** + Art. - **lui**

pl.

N + AC: Stem + Infl. - **e** + Art. - **le**
G + D: Stem + Infl. - **e** + Art. - **lor**

Non - definite article

sg.

N + AC: Stem + Infl. - **e**
G + D: Stem + Infl. - **e**

pl.

N + AC: Stem + Infl. - **e**
G + D: Stem + Infl. - **e**

1.c. FEMININE NOUNS

Rule no. 7

Definite article

sg.

N + AC: Stem + Infl. - **e** + Art. - **a**
G + D: Stem + Infl. - **i** + Art. - **i**

pl.

N + AC: Stem + Infl. - **i** + Art. - **le**
G + D: Stem + Infl. - **i** + Art. - **lor**

Non - definite article

sg.

N + AC: Stem + Infl. - **e**
G + D: Stem + Infl. - **i**

pl.

N + AC: Stem + Infl. - **i**
G + D: Stem + Infl. - **i**

EXCEPTION:

- change of vowel in the stem: G,D sg. / N,G,D,AC pl. - a/ă

Rule no.8

Definite article

sg.

N + AC: Stem + Art. - a

G + D: Stem + Infl. - e + Art. - i

pl.

N + AC: Stem + Infl. - i + Art. - le

G + D: Stem + Infl. - i + Art.- lor

Non - definite article

sg.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - i

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

Rule no. 9

Definite article

sg.

N + AC: Stem + Art. - a

G + D: Stem + Infl. - e + Art. - i

pl.

N + AC: Stem + Infl.- e + Art.- le

G + D: Stem + Infl.- e + Art.- lor

Non - definite article

sg.

N + AC: Stem + Infl. - ă

G + D: Stem + Infl. - e

pl.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - e

EXCEPTION:

- change of vowel in the stem: G,D sg. / N,G,D,AC pl. - **ea/e**

Rule no. 10

Definite article

sg.

N + AC: Stem + Art. - **a**

G + D: Stem + Infl. - **i** + Art. - **i**

pl.

N + AC: Stem + Infl. - **i** + Art. - **le**

G + D: Stem + Infl. - **i** + Art. - **lor**

Non - definite article

sg.

N + AC: Stem + Infl. - **ă**

G + D: Stem + Infl. - **i**

pl.

N + AC: Stem + Infl. - **i**

G + D: Stem + Infl. - **i**

2. ADJECTIVES

Adjective inflexion rules in Romanian also depend on **the type of article** occurring (definite or non-definite) as well as on **the noun - adjective topic** (the adjective precedes the noun or occurs after it). In our implementation we have distinguished among the following rules (where the abbreviations are the same as before):

Rule no. 1

This rule refers to **adjectives with four endings and no phonetical change within the stem**. In what follows it shall be described according to the type of article, the topic and the gender of the adjective. Let us note that, for the Romanian language, in the case of all adjective inflection rules, we have

Neuter SG. = Masculine SG. and Neuter PL. = Feminine PL.

this being the reason for which the neuter gender will not be separately taken into account.

Rule no. 1 can then be described as follows:

1.1. Non - definite article (with the adjective occurring either before or after the noun) + definite article (with the adjective occurring after the noun):

Masculine

sg.

N + AC: Stem

G + D: Stem

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

Feminine

sg.

N + AC: Stem + Infl. - ă

G + D: Stem + Infl. - e

pl.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - e

1.2. Definite article with the adjective occurring before the noun:

Masculine

sg.

N + AC: Stem + Conv. - u + Art. - l

G + D: Stem + Conv. - u + Art. - lui

pl.

N + AC: Stem + Infl. - i + Art. - i

G + D: Stem + Infl. - i + Art. - lor

Feminine

sg.

pl.

N + AC: Stem + Art. - a

N + AC: Stem + Infl. - e + Art. - le

G + D: Stem + Infl. - e + Art. - i

G + D: Stem + Infl. - e + Art. - lor

EXCEPTIONS:

- for adjectives with the stem ending in consonant **t**, the same stem ends in **ŧ** for **masculine plural** - phonetic (consonant) change within the stem in masculine plural (both for 1.1. and for 1.2.): **t / ŧ**; stem ending in **ŧ** + Infl. - i;
- for adjectives with the stem ending in consonant **s**, the same stem ends in **ş** for **masculine plural** - phonetic change within the stem in masculine plural: **s/ş**; stem ending in **ş** + Infl. - i;
- for adjectives with the stem ending in consonant **d**, the same stem ends in **z** for **masculine plural** - phonetic change within the stem in masculine plural: **d/z**; stem ending in **z** + Infl. - i;
- for adjectives having vowel **o** in the last syllable, phonetic (vowel) change within the stem for **feminine singular and plural**: **o/oa**;
- for adjectives having vowel **ă** in the last syllable, phonetic change within the stem for **masculine and feminine plural**: **ă/e**

Rule no. 2

This rule refers to **adjectives with four endings**, having the stem ending in a vowel, and having a **different termination in masculine singular**. It can be described, according to the same criteria, as follows:

2.1. Non - definite article (with the adjective occurring either before or after the noun) + definite article (with the adjective occurring after the noun):

Masculine

sg.

N + AC: Stem + Infl. - **u**

G + D: Stem + Infl. - **u**

pl.

N + AC: Stem + Infl. - **i**

G + D: Stem + Infl. - **i**

Feminine

sg.

N + AC: Stem + Infl. - **ă**

G + D: Stem + Infl. - **e**

pl.

N + AC: Stem + Infl. - **e**

G + D: Stem + Infl. - **e**

2.2. Definite article with the adjective occurring before the noun:

Masculine

sg.

N + AC: Stem + Infl. - **u** + Art. - **l**

G + D: Stem + Infl. - **u** + Art. - **lui**

pl.

N + AC: Stem + Infl. - **i** + Art. - **i**

G + D: Stem + Infl. - **i** + Art. - **lor**

Feminine

sg.

N + AC: Stem + Art. - **a**

G + D: Stem + Infl. - **e** + Art. - **i**

pl.

N + AC: Stem + Infl. - **e** + Art. - **le**

G + D: Stem + Infl. - **e** + Art. - **lor**

Rule no. 3

This rule refers to **adjectives with three endings and having the stem ending in a consonant**; because they contain vowel “o” in the last syllable, they have the phonetic change o/oa in feminine sg. and pl.. The rule can be described, according to the same criteria, as follows:

3.1. Non - definite article (with the adjective occurring either before or after the noun) + definite article (with the adjective occurring after the noun):

Masculine

sg.

N + AC: Stem

G + D: Stem

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

Feminine

sg.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - e

pl.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - e

3.2. Definite article with the adjective occurring before the noun:

Masculine

sg.

N + AC: Stem + Conv. - u + Art. - l

G + D: Stem + Conv. - u + Art. - lui

pl.

N + AC: Stem + Infl. - i + Art. - i

G + D: Stem + Infl. - i + Art. - lor

Feminine

sg.

N + AC: Stem + Infl. - e + Art. - a

G + D: Stem + Infl. - e + Art. - i

pl.

N + AC: Stem + Infl. - e + Art. - le

G + D: Stem + Infl. - e + Art. - lor

Rule no. 4

This rule also refers to **adjectives having three endings**. However, within this class, **the stem ends in a vowel** and the distribution of the terminations is different.

The rule can be described, according to the same criteria, as follows:

4.1. Non - definite article (with the adjective occurring either before or after the noun) + definite article (with the adjective occurring after the noun):

Masculine

sg.

N + AC: Stem + Infl. - u

G + D: Stem + Infl. - u

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

Feminine

sg.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - i

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

4.2. Definite article with the adjective occurring before the noun:

Masculine

sg.

N + AC: Stem + Infl. - **u** + Art. - **l**

G + D: Stem + Infl. - **u** + Art. - **lui**

pl.

N + AC: Stem + Infl. - **i** + Art. - **i**

G + D: Stem + Infl. - **i** + Art. - **lor**

Feminine

sg.

N + AC: Stem + Art. - **a**

G + D: Stem + Infl. - **e** + Art. - **i**

pl.

N + AC: Stem + Infl. - **i** + Art. - **le**

G + D: Stem + Infl. - **i** + Art. - **lor**

Rule no. 5

This rule also refers to **adjectives having three endings**. The stem ends in a **consonant** but the distribution of the terminations is different than that of rule no. 3. The rule can be described, according to the same criteria, as follows:

5.1. Non-definite article (with the adjective occurring either before or after the noun) + definite article (with the adjective occurring after the noun):

Masculine

sg.

N + AC: Stem

G + D: Stem

pl.

N + AC: Stem + Infl. - **i**

G + D: Stem + Infl. - **i**

Feminine

sg.

N + AC: Stem + Infl. - **ă**

G + D: Stem + Infl. - **i**

pl.

N + AC: Stem + Infl.- **i**

G + D: Stem + Infl.- **i**

5.2. Definite article with the adjective occurring before the noun:

Masculine

sg.

N + AC: Stem + Conv. - **u** + Art. - **l**

G + D: Stem + Conv. - **u** + Art. - **lui**

pl.

N + AC: Stem + Infl.- **i** + Art. - **i**

G + D: Stem + Infl.- **i** + Art.-**lor**

Feminine

sg.

N + AC: Stem + Art. - **a**

G + D: Stem + Infl. - **i** + Art. - **i**

pl.

N + AC: Stem + Infl.- **i** + Art.- **le**

G + D: Stem + Infl.- **i** + Art.- **lor**

EXCEPTION:

- phonetic change within the stem in feminine sg., N +AC: **e/ea**

Rule no. 6

This rule refers to **adjectives having two endings and the stem ending in a consonant**. It can be described, according to the same criteria, as follows:

6.1. Non - definite article (with the adjective occurring either before or after the noun) + definite article (with the adjective occurring after the noun):

Masculine

sg.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - e

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

Feminine

sg.

N + AC: Stem + Infl. - e

G + D: Stem + Infl. - i

pl.

N + AC: Stem + Infl. - i

G + D: Stem + Infl. - i

6.2. Definite article with the adjective occurring before the noun:

Masculine

sg.

N + AC: Stem + Infl. - e + Art. - le

G + D: Stem + Infl. - e + Art. - lui

pl.

N + AC: Stem + Infl.- i + Art.- i

G + D: Stem + Infl.- i + Art.- lor

Feminine

sg.

N + AC: Stem + Infl. - e + Art. - a

G + D: Stem + Infl. - i + Art. - i

pl.

N + AC: Stem + Term - i + Art. - le

G + D: Stem + Infl.- i + Art. - lor

Rule no. 7

This rule refers to adjectives with only one ending - **invariant adjectives**. For all cases they consist only of a stem, ending in a vowel.

References:

1. Gramatica limbii române (vol. I). Ediția a doua revăzută și adăugită. București, Editura Academiei, 1966, pp.41-134 (in Romanian).
2. Theodor Hristea (coordinator), Sinteze de limba română. Ediția a treia revăzută și din nou îmbogățită. București, Editura “Albatros”, 1984, pp. 67-70; 203-224 (in Romanian).
3. Luciana Peev, Lidia Bibolar, Jodal Endre, “A Formalization Model of the Romanian Morphology”, in Recent Advances in Romanian Language Technology (editors Dan Tufiş and Poul Andersen). Editura Academiei, București, 1997.

Premises of a Morphological Dictionary of Romanian

Emil Ionescu

1. The Aim

The aim of the present enterprise is the construction of a dictionary of annotated inflected word forms for Romanian. The dictionary (the file *Romdict.txt*) is expressed in the DELAF format (INTEX, Zilberstein 1993) and it is associated with another file containing the description of the labels used in characterizing the inflected forms (*Definitions.txt*). Both files serve as resources for the analyzer constructed by the colleagues from Bulgaria. The analyzer is available at the address <http://www.larflast.bas.bg/balric/tag/default.htm>

The dictionary is a full-form lexicon. It is therefore based on morphological features. It is not 'inflectionally-oriented' (because it does not deal with matters of the inner structure of the word), nor is it based on principles of derivational morphology¹. The main concern was to capture the properties of word forms in terms of features specific to Romanian morphology. This orientation is worth noticing, because it is representative for the present-day trend in HLT: a trend which takes into account the relation between texts and the ultimate building blocks of them, the words.

2. The General Structure of the Dictionary

An entry in this dictionary has the following structure:

- Word form
- Lemma
- Characteristics: lemma characteristics, word form characteristics

A sample:

Word form= **agresivă** ('aggressive' singular, feminine, nominative or accusative)

Lemma= **agresiv**

Characteristics: Lemma characteristics: A (=adjective)+ GR (=gradable)

Word form characteristics: ufsr (=undefinite, feminine, singular, nominative or accusative)

¹ For other tools and resources concerning the morphology of Romanian, see Tufis, L. Diaconu, Barbu and C. Diaconu (1996), Tufis, L. Diaconu, C. Diaconu and Barbu (1996), Peev, Bibolar and Endre (1996), Curteanu, Holban, I. Pavaloi, C. Pavaloi., Negulescu and Todirascu (1996), Vuscan (1996) – all the above mentioned contributions being papers in Dan Tufis (ed.) "Limba și Tehnologie" (Language and Technology), Editura Academiei Române, București, 1996.

The dictionary comprises only synthetic forms. The analytic ones are considered collocations. Synthetic forms in turn may be simple or compound; **agresivă**, for instance, is a simple synthetic form but **nici_un** (the determiner **no**) is compound. Compound forms are written with underscore.

In its present form, the dictionary contains 6768 entries, which cover all the parts of speech of Romanian.

3. The Phases of the Research

The main steps which led the research to the previously mentioned results have been the following:

- Establishing the corpus of the dictionary
- Identifying the lexical forms which belong to the corpus
- Building up the inflectional paradigms (wherever needed)
- Establishing the morphological features involved in the characterization of lemmas and word forms.

3.1. The Corpus

The corpus is composed of a set of 13 articles collected from one of the most important Romanian newspapers, “Evenimentul Zilei” (The Daily Event). It is a representative corpus for the present-day standard Romanian.

3.2. Identifying Lexical Forms

The newspaper articles supplied a number of 1478 word forms, which were isolated through tokenization. Tokenization was performed with a tokenizer built up within the MULTEX Project developed at the Romanian Academy Center of Artificial Intelligence (RACAI), available at www.racai.ro. A sample of a tokenized text is given below:

#el #(he)
#a# (has)
#afirmat# (stated)
#că# (that)
#singura# (The only)
#soluție# (solution)
#pentru_ca # (for)
#populația# (the population)
#să #(to)
#-și# (its own)
#poată #(be able)
#achita# (to pay)
#facturile# (the bills)

#este# (is)
 #creșterea# (the augmentation)
 #reală# (real)
 #a# (of)
 #salariilor# (the salaries)

The word forms identified by tokenization represent 523 lemmas covering both inflectional and non-inflectional parts of speech of Romanian.

3.3. Inflectional Paradigms

For every lemma representing an inflectional part of speech, the corresponding inflection paradigm was built up. This operation has been performed in a manual way. The result was the production of 6768 word forms, which constitute the present form of the dictionary.

3.4 Morphological Features

3.4.1. Establishing the set of morphological features for Romanian represented the linguistic part of the research. In this respect, the way the Bulgarian dictionary was constructed has been followed. Accordingly, a feature was considered morphologically relevant, if that feature was found important for the production and/or distinction of the paradigm members. For instance, the feature A(auxiliary) is in Romanian a *syntactic* feature of the verbs. Nevertheless, it also counts as a morphological one, because the paradigms of the auxiliary verbs **a avea** (to have) and **a vrea** (to want) are distinct from the paradigms of the corresponding 'main' verbs **a avea** and **a vrea**. Likewise, in the case of adjectives, the gender – a semantic feature – has to be equally considered morphologically relevant – just like in Bulgarian, but unlike English – because it serves to distinguish between members of the same paradigm².

3.4.2. With these criteria at hand, twelve (main) parts of speech were identified: noun, verb, adjective, determiner, pronoun, numeral, article, adverb, preposition, interjection, particle and abbreviation.

The last two are not usually encountered in morphological descriptions of Romanian, so some explanations are in point. The category of particles contain certain words with special distribution and behavior. It is about the adverb of negation **nu**, the 'conjunction' **să** (which marks the subjunctive mood), and the so-called 'morpheme' **a** (which marks a certain form of infinitive).

As for abbreviations (for instance, **tel**, from **telefon**), the option to adopt them as a distinct category, originated in the fact that, in general, an abbreviation substantively differs from its corresponding 'full' part of speech. For example, **tel** does not present case inflections, nor can it be used in plural. Anyway, given the marginal character of abbreviations

² It is just for this reason that the very important verb feature *transitivity* is not considered here a morphological feature: it contributes in no way to the formation of the verb paradigm in Romanian.

(especially from the quantitative point of view) their unexpected presence in the morphology of Romanian does not rise serious problems.

3.4.3. It may happen that the same feature is counted as a *lemma feature* in relation to a given part of speech but as a *word form feature* in relation to another part of speech. This comes about, for instance, with gender, which characterizes the *lemma* of nouns and the *word form* of adjectives.

3.4.4 There is a slight difference in the way the Bulgarian and the Romanian dictionary deal with homonymous morphological forms. The difference may be illustrated in the treatment of the following example. For the word form **fly**, in the Bulgarian dictionary the analysis is *pr12sg:123pl* (= present tense, 1st or 2nd person, singular, or 1st or 2nd or 3rd person, plural), while in the Romanian one, it is **fly** *pr12sg; fly/123pl*. In the Romanian dictionary the word form is therefore registered twice (if the homonymy is illustrated by two word forms).

3.4.5 The set of morphological features of Romanian looks as follows:

Nouns

Lemma Features

- Noun
- Common or Proper
- Masculine or Feminine or Neuter

Word Form Features

- Singular or Plural
- Definite or Indefinite
- Nominative or Accusative; Genitive or Dative

Verbs

Lemma Features

- Verb
- Auxiliary (default feature)

Word Form Features

- Indicative or Subjunctive or Imperative or Infinitive or Gerund or Participle
- Present or Imperfect or Recent Perfect or Past Perfect
- 1st person or 2nd person or 3rd person
- Singular or Plural

Adjectives

Lemma Features

- Adjective
- Gradable or Non-gradable

Word Form Features

- Definite or Undefinite
- Masculine or Feminine
- Singular or Plural
- Nominative or Accusative or Dative or Genitive

Pronouns

Lemma Features

- Pronoun
- Personal or Reflexive or Demonstrative or Negative or Indefinite or Interro-Relative

Word Form Features

For Personal Pronouns

- Strong or Weak Form
- 1st person or 2nd person or 3rd person
- Singular or Plural
- Nominative or Accusative or Dative or Genitive
- Masculine or Feminine

For Reflexive Pronouns

- Strong or Weak Form
- 1st person or 2nd person or 3rd person
- Singular or Plural
- Nominative or Accusative or Dative

For Demonstrative Pronouns

- Singular or Plural
- Nominative or Accusative or Dative or Genitive

For Negative Pronouns

- Singular or Plural
- Nominative or Accusative or Dative or Genitive

For Interro-Relative Pronouns

- Nominative or Accusative or Dative or Genitive

For Indefinite Pronouns

- Nominative or Accusative or Dative or Genitive

Determiners

Lemma Features

- Determiner
- Demonstrative or Possessive or Negative or Indefinite or Interro-Relative or Emphatic

Word Form Features

For Demonstrative Determiners

- Singular or Plural
- Nominative or Accusative or Dative or Genitive

For Negative Determiners

- Singular or Plural
- Nominative or Accusative or Dative or Genitive

For Indefinite Determiners

- Nominative or Accusative or Dative or Genitive

For Interro-Relative Determiners

- Nominative or Accusative or Dative or Genitive

For Possessive Determiners

- 1st person or 2nd person or 3rd person
- Singular or Plural

For Emphatic Determiners

- 1st person or 2nd person or 3rd person
- Singular or Plural
- Nominative or Accusative or Dative or Genitive

Articles

Lemma Features

- Article
- Demonstrative or Possessive or Indefinite

Word Form Features

- Masculine or Feminine
- Singular or Plural
- Nominative or Accusative or Dative or Genitive

Numerals

Lemma Features

- Numeral
- Cardinal or Ordinal

Word Form Features

- Masculine or Feminine
- Singular or Plural

Adverbs

Lemma Features

- Adverb
- General or Verbal or Interro-Relative

Word Form Features: none

Conjunctions

Lemma Features

- Conjunction
- Coordinating or Subordinating

Word Form Features: none

The RORIC-LING Bulletin

months 13 - 18

A number of **65** questions have been asked, by subscribers coming mostly from Romanian academic units, but not only. Software companies, both from Romania and from abroad, are also represented. Some of these questions have been asked more than once, as will be specified in the bulletin. There are two main categories of questions, corresponding to the topics of the BALRIC-LING Romanian part of the project, as follows:

- general questions concerning the two discussed approaches to morphology;
- specific questions concerning the full-form lexicon and the corresponding implementation.

The questions have been grouped according to the topic they refer to (and not according to the subscribing date, namely not in chronological order). All information concerning the names and personal data of subscribers is stored in the RORIC-LING files but has been deleted from the bulletin, in order to facilitate reading and using this material, as well as the search process according to topic.

Some statistics:

Questions: 65

Country:	Romania	Other*
Questions:	41	24

Native Language:	Romanian	Other
Questions:	52	13

Activity Field:	Education	Research	Software Industry	Other
Questions:	29	17	12	7

* AUSTRALIA, CANADA, GERMANY, ITALY, UNITED KINGDOM, UNITED STATES

General Questions Concerning the Two Discussed Approaches to Morphology

What exactly is a grammatical (or morphological) dictionary? (asked twice)

The grammatical dictionary is a representative summary of all basic word forms in a certain language accompanied by their grammatical characteristics. These features determine the generation of all word forms which are derived from the basic one (its word-formation) and provide the basic information for the results of the text analysis. Grammatical dictionaries are among the first NLP applications and represent a basic tool for collecting and organizing the linguistic data.

The morphological dictionary is a database which provides a wide range of data about the morphological characteristics and the forms of a certain word. It also allows for quick retrieval of grammatical information coming simultaneously from different templates (paradigm tables). The main purpose of a grammatical dictionary is to identify the relations between a concrete word form and its invariant (lemma). The purpose of the morphological dictionary is therefore to identify the word form and its characteristics and to classify it with regard to its lemma.

For more information on grammatical (or morphological) dictionaries, see the Bulgarian page of the BALRIC-LING project, at

http://www.larflast.bas.bg/balric/index/index_eng.htm

What is the difference between "root" and "stem"? (asked twice)

The terms **stem**, **theme** or **thema** designate the **root + affixes/infixes**. Therefore the stem represents the inflexional base of a word to which other elements, such as thematic vowels and consonants, inflections etc. are added. Obviously, in many cases, the stem of a word can be identical to its root. For example, in Greek, the root *lip* underlies the present theme *leip* to which the inflection *ein* is added to form the present infinitive of the verb *léipein* (to leave).

What are the benefits of using the full-form lexicon in comparison to the inflexional approach to morphology? (asked 3 times)

The main benefits are:

1. One avoids once and forever the painful, endless inflexional discussion at morphemic level and focuses directly on the bricks-words (but note, all

word forms are considered in isolation). Inflexion concerns the word-formation language particularities. However, since today HLT has moved to text analysis for most European languages, one should be preoccupied in highlighting the word -> text focuses, a task which can be fulfilled by using the full-form lexicon.

2. Mapping any text onto such a lexicon allows one to start discussing the POS-disambiguation problems, which today are considered the real problems of text analysis (at "word-level").

What is the basic criteria according to which a specific feature is included in the full-form lexicon? (asked 3 times)

The main criteria for inclusion of a feature can be expressed by the following question: "Is the feature to be included important for the production and distinction of paradigm members?"

Specific Questions Concerning the Full-Form Lexicon and the Corresponding Implementation

Can a morphological dictionary be used in order to design a spell checker for Romanian? (asked twice)

We don't think that it can be used directly. However, taking into consideration that, within such a dictionary, all inflexional forms of a word are present, we think that it can serve (when being complete, not as in our case a mere sample) for the creation of a list of words that could, in turn, be used in order to design a spell checker. Such a solution would only make use of a small part of the information which is included in a morphological dictionary.

What's the difference between a full-form lexicon and a derivational/inflectional one? (asked twice)

The main difference is that in a full-form lexicon one doesn't have the representation of the word's structure. You have only the relevant information about the word plus the form as such.

Why did you choose only newspaper articles for your corpus? (asked twice)

Just because the language in newspapers is representative for contemporary Romanian. This is a routine practice in work on corpora.

Isn't transitivity a feature interesting from a morphological point of view in Romanian? What is the reason for not including it into the set of verb features? (asked twice)

The reason is that, from a morphological point of view, transitivity is not relevant in Romanian since transitive verbs do not show special inflected forms.

Do you intend to extend the dictionary? (asked twice)

We would like to, but this depends on the opportunities regarding a new project.

Are there any other (on-line) contributions to a morphological lexicon for Romanian ? (asked twice)

To the best of our knowledge, no. But I also know that different attempts (and ongoing projects) exist in Bucharest (Romanian Academy Institute for Artificial Intelligence) and Cluj.

The passive voice is a morphological or a lexical category in Romanian? Does your lexicon include passive constructions? (asked twice)

Prescriptive grammars view the passive voice as a morphological category. We have serious doubts concerning this point of view. We preferred not to consider the passive voice a morphological category. We don't have passive constructions in our lexicon.

How do you deal with cases of morphological ambiguity? (asked twice)

I am repeating the explanation given in my web presentation. Suppose we have a word form such as **fly**. The lemma features help us to disambiguate the part of speech. So we have two lemmas, one for the noun **fly** and one for the verb **fly**. As for the verb, it is registered twice, with the following information: **fly** pr12sg; **fly**123pl.

It is unclear to me why you make the difference between proper names and common nouns. (asked twice)

This difference is needed because there are morphological differences between proper names and common nouns and our criterion has been the following: treat as morphologically relevant any feature (be it semantic or syntactic) which has morphological (i.e. inflectional) consequences.

Is the adjective in Romanian a category bearing article, too? Please explain to me the difference between an articulated and a non-articled adjective. (asked twice)

Adjectives in Romanian have (definite) article indeed. Roughly speaking, this happens when they are placed in prenominal position. For instance, the adjective **frumos** (nice, beautiful) is non-articled in postnominal position (copilul frumos, literally child-the beautiful, that is "the beautiful child"), but articulated in prenominal position (frumosul copil, literally beautiful-the child).

Wouldn't it possible to enrich your dictionary in a purely automatic way, that is, by inputting a word manually and by further constructing the rest of the paradigm automatically? (asked twice)

Of course it would, the only problem is building up such a program. This remains one of our tasks for the future.

What's the purpose of such a dictionary? (asked twice)

Such a dictionary could serve, for instance, as a tool in second language teaching.

Is your lexicon able to tell me what the inner structure of a word is? (asked twice)

No, it isn't. All it can do is give the word form along with its relevant information (for instance gender, number, case).

As far as I know, Romanian uses analytical ways of expressing the comparison degrees of adjectives. Is this the reason for not specifying the comparison degrees in your dictionary? (asked twice)

Yes. Adjectives with comparison degrees in Romanian are treated as words composed of other words.

Why do you need a distinction between lemma features and word form features? (asked twice)

The difference between lemma features and word form features has been adopted for reasons of uniform description. To a certain extent, it is a theoretical distinction, too, but we believe that a description made without it works equally well.

6678 word forms is a too small sample of a dictionary. Do you intend to further extend the lexicon? (asked twice)

Yes, we do, but this is supposed to be a part of another project.

The category of particles is not convincingly defended in your presentation. Would you like to be more explicit about the reasons for including it in your lexicon? (asked twice)

This category is fairly heterogeneous, indeed, but there is no other way to deal with words that are neither adverbs nor another part of speech. So, the solution we adopted was one of "high emergency".

Are the possessive and the demonstrative articles also represented in your dictionary? I was not able to find such a part of speech. (asked twice)

The two so-called 'articles' do not occur in the lexicon, probably because the corpus does not contain them. But there is no difficulty to extend the lexicon with these categories.

There is something that I am not yet able to understand, as far as the relation between the corpus and the dictionary is concerned. Does the lexicon contain only the word forms already contained in the corpus? Or does it contain more, that is, the full paradigm represented in the corpus by, say, one or two members? (asked twice)

It's very easy to check out this relationship (provided, of course, that you know Romanian a little bit!). But to be more specific, I will tell you that the lexicon is richer than the corpus. The corpus contains about 1500 word forms, while the lexicon contains the full paradigm of a word form occurring in the corpus.

The same feature is alternatively registered as a lemma feature and a word form feature. Why?

This is because, in one case it only characterizes the word form, while in the other case it is specific to the lemma itself. For instance, the gender of the noun is a lemma feature, because it does not determine the inflection. Nevertheless, the gender of the adjective does determine the inflection, and so it is a word form feature.

How does your tokenizer treat the Romanian word / construction *am dormit*? As two different words or as a single word?

Am dormit is taken to be a compound word - a collocation. This is the analysis provided for all compound verbal forms.

Do the words which are analyzed morphologically exist in a dictionary or are they analyzed automatically?

I'm not sure I understand what you mean. If you are referring to the words you can find in the dictionary, they are there along with the relevant (morphological) information. But if you are referring to the way this information has been assigned to the word form, I have to say that this has been done in an automatic way.

Is the extension of the lexicon performed in an automatic way or in a manual one?

The extension of the lexicon has been performed manually.

How many members are there in the case system of Romanian?

Leaving aside the vocative, there are four cases in Romanian: nominative, accusative, genitive, and dative.

I tried to access the page of the tokenizer and found nothing there. Did the address change in the meantime?

As far as I know, the address is the same. Try again!

What is the utility of the morphological analyzer?

The analyzer provides the information required in connection with a given word form.

Is the tokenizer language-independent ?

The tokenizer is language-independent in the sense that, if you give it a training corpus from a language different from Romanian, it will be able to perform the same task as the one for Romanian.

What is the utility of a tokenizer in text processing?

The tokenizer helps you to extract lexical items from a text faster and easier than in the manual way.

How do you analyze compound words?

A compound word is considered a single lexical item, however composed of other words. We mark compound words with underscore: **nici_un** (no one).

Why do you maintain the distinction between lemma features and word form features?

This is because, in one case a feature only characterizes the word form, while in another case it is specific to the lemma itself. For instance, the gender of the noun is a lemma feature, because it does not determine the inflection. Nevertheless, the gender of the adjective does determine the inflection, so it is a word form feature.

You work with the distinction articulated / non-articled noun, but as for the articulated ones you leave aside the distinction definite / non-definite. Why?

Good question! We have to incorporate this pair of features, too, because it determines the inflection of the nouns.

Lucrările Centrului Regional de Informare în Tehnologia Limbajului din România

Această carte conține toate lucrările ale căror autori sunt membri ai Centrului Regional de Informare românesc din domeniul tehnologiei limbajului (RORIC-LING), împreună cu eșantioane de date relevante și cu buletinele corespunzătoare celor trei seminarii virtuale, care au fost ținute la fiecare șase luni (2001-2003). Ea reprezintă o încercare de a contribui la creșterea gradului de informare asupra unora dintre cele mai avansate tehnologii privitoare la limbajul natural, ca și asupra posibilelor aplicații de natură științifică și industrială ale resurselor lingvistice corespunzătoare atât în România, cât și în întreaga zonă a Balcanilor.

RORIC-LING este o parte a proiectului mai larg BALRIC-LING, proiect finanțat de Comisia Europeană (IST-2000-26454). Scopul RORIC-LING este acela de a mări gradul de informare referitoare la HLT (Human Language Technologies) în primul rând în România, o țară în care piețele interesate în HLT, ca și adevăratele aplicații de inginerie a limbajului lipsesc încă. Biroul de informare RORIC-LING a fost deschis atât specialiștilor, cât și nespecialiștilor, adresându-se, în egală măsură, unor participanți la seminarii virtuale, care proveneau din mediul academic, din cel al cercetării, de la companii de software, precum și din alte sfere de activitate.

Întrucât HLT este un domeniu suficient de vast, RORIC-LING abordează numai următoarele trei teme:

- formalisme gramaticale și utilizarea lor în cazul limbii române; instrumente corespunzătoare pentru adnotarea corpusurilor;
- generarea semiautomată a synset-urilor și cluster-elor românești de tip WordNet;
- o specificație teoretică pentru un model morfologic al limbii române.

Lucrarea de față reflectă situl web¹ creat în cadrul acestui proiect, pentru a facilita comunicarea cu potențialii clienți, precum și răspândirea rapidă a informațiilor privitoare la tematica RORIC-LING. Acest sit web conține toate lucrările specialiștilor RORIC-LING, însoțite de eșantioane de date corespunzătoare și demonstrații on-line, împreună cu buletinele care reflectă seminariile virtuale ce au avut loc cu privire la fiecare temă abordată de RORIC-LING.

Situl proiectului ca atare este mult mai bogat în informații decât lucrarea de față. El include ample sinteze, ale căror autori sunt specialiști de la ILSP (Grecia) și Sheffield University (Anglia) și care se referă la principalele teme abordate în cadrul proiectului BALRIC-LING. Același sit este extrem de bogat în demonstrații on-line, a căror utilizare o recomandăm tuturor vizitatorilor care doresc să se familiarizeze cu un domeniu de cercetare relativ nou.

În cadrul acestei cărți tematica RORIC-LING este tratată bilingv ca, de altfel, și în pagina web a proiectului. Prima parte a cărții este în limba engleză, iar cea de-a doua parte reprezintă traducerea corespunzătoare în românește. Fiecare dintre cele două părți identice este organizată conform tematicii RORIC-LING. Toate materialele prezentate aici au fost preluate direct de pe situl web al proiectului. Ca rezultat al unui efort minim de editare, impus de durata scurtă a proiectului, precum și pentru ca această carte să reflecte cât mai fidel pagina web a proiectului, stilul ei nu este unitar. Rugăm cititorul să accepte scuzele de rigoare pentru acest inconvenient.

Sperăm că lucrarea de față va suscita interesul tuturor celor implicați în domeniul tehnologiei limbajului, dar și al celor care nu sunt încă familiarizați cu acest domeniu relativ nou, mai ales în România. Principalul scop al acestei cărți este acela de a se constitui într-o invitație deschisă, adresată tuturor cititorilor ei, de a accesa pagina web a proiectului, care va avea, desigur, mult mai multe de oferit.

Echipa RORIC-LING este profund îndatorată Comisiei Europene pentru a fi încurajat acest efort de creștere a gradului de informare. Dorim să mulțumim

¹ <http://phobos.cs.unibuc.ro/roric>

Comisiei Europene pentru importanța pe care a acordat-o tematicii abordate în cadrul RORIC-LING, precum și pentru întregul sprijin oferit. Mulțumiri speciale datorăm, de asemenea, Dr. Galia Angelova, de la Academia Bulgară de Științe, Linguistic Modelling Department, coordonatorul proiectului BALRIC-LING, pentru continua susținere acordată echipei noastre atât pe durata întregului proiect, cât și pe parcursul ultimilor ani.

Februarie 2003

Florentina Hristea

Marius Popescu

Obiectivele proiectului BALRIC-LING

Dr. Galia Angelova

Coordonator BALRIC-LING

RORIC-LING este o parte a proiectului mai larg BALRIC-LING, finanțat de Comisia Europeană (IST-2000-26454).

Primul dintre obiectivele principale ale proiectului BALRIC-LING este acela de a mări gradul de informare asupra potențialului celor mai avansate tehnologii privitoare la limbajul natural, ca și asupra posibilelor aplicații de natură științifică și industrială ale resurselor lingvistice corespunzătoare, în special în noile țări asociate din zona Balcanilor, Bulgaria și România. Creșterea gradului de informare privitoare la HLT (Human Language Technologies) este extrem de importantă în aceste țări, întrucât ele se află pe drumul integrării depline în Comunitatea Europeană, dar exemplele de aplicații comerciale de succes de tip HLT sunt inexistente pentru limbile română și bulgară. Datorită izolării prelungite față de vastul schimb științific de idei și metode practice privitoare la ingineria limbajului dintre țările Europei Occidentale și fiind confruntate cu studiul unor limbi structural diferite, puținele grupuri avansate de cercetători în domeniul HLT din ambele țări nu pot încă să creeze, exclusiv prin eforturi proprii, o masă critică de specialiști informați, care să poată mări calitatea aplicațiilor din domeniul ingineriei limbajului pentru noile piețe interesate în HLT, care acum se nasc, atât în Bulgaria, cât și în România.

Întrucât HLT reprezintă un domeniu extrem de vast, BALRIC-LING se concentrează asupra a patru teme principale:

resurse lingvistice și adnotări centrate în jurul cuvântului;

corpusuri și tagging;

instrumente relevante pentru tratarea și realizarea acestora;

posibile utilizări avansate de tip HLT ale resurselor luate în considerație.

Pentru a-și atinge scopul de creștere a gradului de informare privitor la acestea în Bulgaria și în România, BALRIC-LING și-a propus realizarea inițiativelor principale la care ne vom referi în cele ce urmează.

O primă inițiativă este aceea a dezvoltării unor centre regionale de informare ("Regional Information Centers" - RIC) în BULgaria și ROMania (BULRIC și respectiv RORIC). Aceste centre de informare reprezintă situri Web conținând descrieri ale unor instrumente de tip HLT, eșantioane de date, resurse lingvistice și prototipuri ale unor instrumente corespunzătoare. Siturile susțin câte un birou de informare în cadrul căruia specialiști ai consorțiului au pregătit sinteze amănunțite privitoare la cele patru teme ale proiectului. Tot aici persoanele și instituțiile interesate au putut să își adreseze întrebările legate de tematica fiecărui centru de informare în parte. Pe aceste situri sunt disponibile materiale în limba engleză și, în cazul fiecărei țări în parte, în bulgară, respectiv în română, întrucât, numai în acest mod, informația respectivă poate deveni accesibilă publicului larg, precum și grupurilor de cercetători și companiilor interesate în domeniu, din ambele țări. În plus, siturile centrelor de informare conțin informații cu privire la conferințele din domeniu, școlile de vară și workshop-urile organizate în Europa și strâns legate de tematica BALRIC-LING. Acest tip de informație este extrem de utilă, întrucât știrile referitoare la astfel de evenimente sunt extrem de rare în Bulgaria și în România.

Seminarii virtuale bazate pe întrebările primite la birourile de informare ale RIC-urilor au fost organizate la fiecare șase luni. Persoanele care au subscris unor liste de corespondență cu largă acoperire în domeniu și organizate în mod special cu acest scop, au putut pune întrebări privitoare la toate materialele prezentate de către centrele de informare corespunzătoare. Specialiști ai consorțiului au formulat răspunsuri care au fost expediate tuturor celor care au subscris. Aceste seminarii virtuale au facilitat sporirea gradului de cunoaștere și informare, precum și distribuirea expertizei de la instituțiile academice mai informate către organizațiile industriale interesate din Bulgaria și România. Buletine virtuale bianuale în limbile

bulgară și respectiv română au fost direcționate către participanții la seminarul virtual și au permis diseminarea pe scară largă a inițiativelor BALRIC-LING, în special printre instituțiile lingvistice și în rândul persoanelor aflate în provincie, atât în Bulgaria, cât și în România.

Cel de-al doilea dintre obiectivele principale ale proiectului BALRIC-LING este acela de a ajuta grupurile de cercetători din Balcani să devină mai bine pregătite pentru cooperarea științifică la nivel european. ILSP (Grecia) și Universitatea Sheffield (UK) și-au oferit bogata experiență practică în realizarea cercetării științifice de succes, atât la scară națională, cât și europeană. Centrele regionale de informare din Bulgaria și din România au contribuit în mod esențial la diseminarea ideilor și conceptelor HLT, inclusiv printre firmele de soft interesate în dezvoltarea ulterioară a unor aplicații avansate de tip HLT pentru limbile bulgară și respectiv română.

Una dintre modalitățile de facilitare a formării consorțiilor pentru realizarea unor proiecte științifice viitoare o reprezintă schimbul de informații cu privire la formatele existente și la standardizarea reprezentării interne ale unor resurse aparținând membrilor proiectului. Fiind pregătite în formate unitare, aceste resurse pot fi ușor integrate pentru utilizarea simultană în aplicații multilingve și pot fi dezvoltate în continuare. Toate cerințele de standardizare stabilite în cadrul BALRIC-LING au fost făcute publice, pentru ca grupurile de cercetători din Balcani și firmele de software interesate să poată avea acces la ele și să le poată folosi ca repere.

BALRIC-LING își propune standardizarea a două formate ale reprezentării interne:

standardizarea formatelor pentru codificarea corpusurilor monolingve și paralele;
standardizarea formatelor pentru reprezentarea internă a dicționarelor morfologice în cele trei țări balcanice.

Configurația BALRIC-LING compactă permite familiarizarea în profunzime cu toate detaliile transmise prin strânsa comunicare dintre parteneri. Participanții BALRIC-LING din țările membre ale UE vor pregăti un referat de evaluare a progresului realizat în cele două țări asociate din Balcani, Bulgaria și România, cu privire la sarcinile de ridicare a gradului de cunoaștere și de diseminare a informației, pe care și le-au propus.

I

FORMALISME GRAMATICALE ȘI UTILIZAREA LOR ÎN CAZUL LIMBII ROMÂNE. INSTRUMENTE CORESPUNZĂTOARE PENTRU ADNOTAREA CORPUSURILOR

Gramatici de dependență și gramatici WG

Florentina Hristea și Marius Popescu

În ciuda diverselor teorii lingvistice existente, care au ca rezultat modalități total diferite de a privi structura propoziției și, prin urmare, procesul de analiză sintactică, majoritatea lingviștilor sunt astăzi de acord cu faptul că în centrul conceptului de **structură a propoziției** se află **relațiile dintre cuvinte**. Aceste relații pot fi de diverse naturi, referindu-se ori la funcții gramaticale (subiect, complement etc.), ori la acele legături care îmbină cuvintele în unități mai largi, cum ar fi grupurile sintactice sau chiar propozițiile în ansamblul lor.

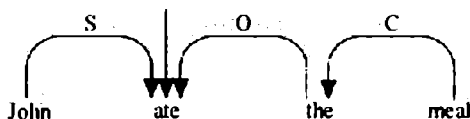
Spre deosebire de gramaticile generative, **gramaticile de dependență** (Mel'cuk 1962, 1963, 1964 până la Mel'cuk 1979, 1987 și 1988), precum și **gramaticile WG** (abreviere de la "*word grammar*", Hudson 1984, 1990 și 1998) **nu se bazează pe noțiunea de constituent ci pe relațiile directe existente între cuvinte**, privite fiind ca niște relații de dependență. Gramaticile WG reprezintă o variantă a gramaticilor de dependență și, în același timp, o teorie care a fost cu succes adaptată și aplicată specificului limbii române [1], [2]. Caracteristica distinctivă principală a acestui tip de gramatici o constituie utilizarea de către acestea a dependențelor în defavoarea structurii bazate pe constituenți.

Teoria gramaticilor de dependență necesită punerea în evidență a unei *structuri de dependență*, care poate fi privită, printre altele, ca opunându-se structurii alcătuite din constituenți. Așa cum caracterizează Richard Hudson aceste structuri, în timp ce structura dată prin constituenți a unei propoziții "constă din relații de la parte la întreg între cuvinte și grupurile sintactice", structura de dependență a unei propoziții "constă din relații de la cuvânt la cuvânt între cuvinte individuale". Ideea centrală pe care se bazează noțiunea de dependență este aceea că fiecare cuvânt este privit ca depinzând de cuvântul care îl leagă de restul propoziției, practic explicând de ce este utilizat. Acesta din urmă este numit *capul* celuilalt. Structura sintactică a propoziției este reprezentată prin relații sintactice binare între cuvinte. În cadrul fiecărei perechi de cuvinte aflate într-o relație de acest tip, unul dintre cuvinte (cel *dependent*) depinde de celălalt (*capul*), care îl susține atât din punct de vedere sintactic, cât și semantic. În același timp, *valența* unui cuvânt este considerată a fi mulțimea dependențelor în care acesta poate să fie implicat: subiectul său, complementele sale etc.

La baza gramaticilor de dependență se află relația dintre cuvântul cap și cuvântul dependent. **Analiza sintactică a unei propoziții** înseamnă, din punctul de vedere al gramaticilor de dependență, descrierea tuturor relațiilor de dependență care intervin între toate cuvintele unei propoziții. O caracteristică importantă a acestui tip de analiză sintactică este aceea că ea *tratează grupurile sintactice ca*

reprezentând produse secundare ale dependențelor stabilite. O analiză sintactică de acest tip a unei propoziții poate fi descrisă prin intermediul unei diagrame relativ simple care pune în evidență unitățile componente (fiecare cuvânt în parte) și relațiile ce se stabilesc între ele. Pe scurt, teoria generală consideră toate unitățile sintaxei ca fiind cuvinte individuale.

Întrucât analiza sintactică bazată pe constituenți, adică utilizând gramatici PS (de la "phrase-structure grammar"), cu care încercăm să facem o comparație, este tipică limbii engleze, să considerăm analiza sintactică de dependență (care utilizează gramatici de dependență) a următoarei propoziții englezești:



Propoziția (*El a luat masa*) conține patru cuvinte, astfel încât diagrama corespunzătoare ia în considerație patru unități și pune în evidență relațiile care se stabilesc între ele. Relațiile sunt reprezentate prin săgeți, fiecare dintre acestea ținând de la cuvântul dependent spre cuvântul cap. Etichetele indică tipul relației ("s" pentru subiect, "o" pentru obiect și "c" pentru complement, altul decât cel direct). Săgeata verticală indică unicul cuvânt care nu depinde de nici un altul, atât din punct de vedere sintactic, cât și semantic (verbul "ate"). Faptul că "the meal" constituie un grup sintactic (alcătuit dintr-un determinant și un substantiv) este indicat prin săgeata care leagă cuvântul "the" direct de cuvântul "meal".

Se remarcă faptul că, într-o analiză sintactică bazată pe constituenți, grupul sintactic "the meal" ar reprezenta o realizare a regulii de rescieri

NP → Det N

și, prin urmare, ar reprezenta un grup nominal organizat în jurul substantivului "meal" cu rol de cap sau centru al grupului sintactic. Într-o analiză sintactică bazată pe gramatici de dependență, în schimb, cuvântul "the" reprezintă capul, în timp ce substantivul "meal" este cuvântul dependent.

În efectuarea unei **analize sintactice de dependență** prin intermediul unei diagrame de acest tip, trebuie avut permanent în vedere faptul că funcția gramaticală este întotdeauna funcția cuvântului dependent în relația sa cu capul. Așa cum s-a menționat deja, ideea care stă la baza acestei diagrame este aceea că fiecare cuvânt depinde de cuvântul care îl leagă de restul propoziției, explicând în mod implicit de ce el este utilizat. Conceptul central pe care se bazează gramaticile de dependență este acela al *relațiilor de dependență existente între cuvinte individuale*. În acest cadru, **analiza sintactică** înseamnă descrierea structurii unei întregi propoziții ca fiind o mulțime de relații de dependență existente între perechi

de cuvinte individuale ale acelei propoziții. Paragraful următor va oferi o definiție mai riguroasă a *structurii sintactice a unei propoziții*, prin care vom înțelege o structură (S,D), unde S va desemna *structura de dependență*, iar D va desemna *tipul dependențelor*.

O întrebare care se ridică în mod natural este cea care încearcă să elucideze de ce și când ar putea fi preferat acest formalism pentru descrierea structurii sintactice a propozițiilor limbajului natural. Dintre numeroasele avantaje oferite de formalismul generat de gramaticile de dependență amintim aici doar faptul că acesta poate descrie fenomene lingvistice cum ar fi *existența constituenților discontinui* sau *variația ordinii cuvintelor* în cadrul propoziției. Astfel, în timp ce limbi cum ar fi rusa sau latina, precum și majoritatea limbilor romanice, prezintă o ordine a cuvintelor extrem de flexibilă (chiar dacă nu arbitrară), care însă poate fi surprinsă și modelată cu ajutorul acestui formalism, limba engleză se caracterizează printr-o ordine a cuvintelor mai degrabă rigidă. Nu este, prin urmare, întâmplător faptul că analiza sintactică de tip PS își are originile în Statele Unite și că ea s-a dezvoltat cu specială referire la limba engleză.

1. Relații de dependență

Atunci când nu sunt specificate nici un fel de restricții, între cuvintele unei propoziții poate exista o întreagă varietate de relații de dependență. Rolul gramaticilor de dependență este în primul rând acela de a specifica restricțiile pe care relațiile de dependență trebuie să le satisfacă astfel încât structura pe care ele o definesc să fie corectă din punct de vedere lingvistic.

Indiferent de limba la care se referă, orice gramatică de dependență trebuie să ia în considerație următoarele trei *principii generale*, motivate din punct de vedere lingvistic:

- orice cuvânt trebuie să depindă de un singur alt cuvânt (capul), cu excepția predicatului propoziției, care nu depinde de nici un alt cuvânt;
- mai multe cuvinte pot depinde de același cap;
- dacă relațiile dintre cuvintele dependent și cap sunt reprezentate prin arce orientate de la dependent spre cap, atunci aceste arce nu trebuie să se intersecteze, iar graful orientat astfel format trebuie să fie aciclic.

Dincolo de respectarea acestor principii generale, o gramatică de dependență mai poate, de asemenea, specifica ce relații sunt admise între diverse cuvinte, conform părții de vorbire pe care acestea o reprezintă sau conform altor criterii. Spre exemplu, o gramatică de dependență ar putea stipula faptul că un verb nu poate depinde de un substantiv sau ar putea indica de care părți de vorbire poate depinde un adjectiv etc.

Relațiile de dependență care definesc structura unei propoziții pot fi descrise prin intermediul *structurii de dependență* și al *tipului* dependențelor. *Structura de*

dependență va specifica, în cazul fiecărui cuvânt, care sunt celelalte cuvinte de care acesta depinde. *Tipul dependențelor* va specifica, în cazul fiecărei dependențe, tipul acesteia.

Din punct de vedere formal, relațiile de dependență pot fi descrise după cum urmează:

Fie W o mulțime finită de *cuvinte* (vocabulary) cu ajutorul căroră se formează propozițiile. O propoziție va fi desemnată prin secvența de cuvinte

$$w_1, w_2, \dots, w_n$$

Vom nota prin w_0 un cuvânt special numit *BOS* (de la "beginning of sentence"), care denotă începutul propoziției și prin w_{n+1} cuvântul special *EOS* (de la "end of sentence"), care denotă sfârșitul propoziției.

Fie T o mulțime finită de *etichete* numite părți de vorbire. Acestea vor desemna părțile de vorbire (substantiv, verb, adjectiv etc.) căroră le pot aparține cuvintele vocabularului W .

Fie D o mulțime finită de *tipuri ale dependențelor* (subiect, atribut, determinant etc.).

Structura sintactică a unei propoziții

$$w_1, w_2, \dots, w_n$$

va fi o structură (S, D) , unde prin S vom desemna *structura de dependență*, iar prin D vom desemna *tipul dependențelor*.

Structura de dependență S este, la rândul ei, o structură (T, P) , unde $T = t_1, t_2, \dots, t_n; t_i \in T \forall 1 \leq i \leq n$. T este o secvență de etichete t_i ce desemnează părțile de vorbire căroră le aparține fiecare cuvânt w_i . În cadrul aceleiași structuri de dependență, $P = p_1, p_2, \dots, p_n; p_i \in \{1, 2, \dots, n+1\} \forall 1 \leq i \leq n$ este o secvență de numere care specifică, corespunzător fiecărui cuvânt w_i , capul său (cuvântul de care acesta depinde). Capul sau părintele cuvântului w_i va fi cuvântul w_{p_i} . Întrucât există un cuvânt w_n care nu depinde de nici un altul, se va presupune că părintele acestui cuvânt este $w_{n+1} = EOS$ și că, prin urmare, $p_n = n+1$.

Tipul dependențelor D este o funcție $D: \{1, 2, \dots, n\} \rightarrow D$, unde $D(i) = d$ reprezintă tipul dependenței dintre cuvintele w_i și w_{p_i} .

Principiile generale de care relațiile de dependență trebuie să țină cont și pe care trebuie să le îndeplinească pot fi "traduse", conform acestui formalism, după cum urmează: pentru orice propoziție

$$w_1, w_2, \dots, w_n$$

- $\exists h, 1 \leq h \leq n$ a.î. $p_h = n+1$ și $\forall i, 1 \leq i \leq n, i \neq h, p_i \in \{1, 2, \dots, n\}$ și $p_i \neq i$.
- $\forall 1 \leq i < j \leq n$
 - dacă $p_i < i$, atunci $p_j \leq p_i$ sau $p_j \geq i$;
 - dacă $i < p_i \leq j$, atunci $p_j \leq i$ sau $p_j \geq p_i$;
 - dacă $p_i > j$, atunci $i \leq p_j \leq p_i$.

(Condițiile mai sus amintite reprezintă exprimarea formală a faptului că arcele - sau săgețile - definite de relațiile de dependență nu trebuie să se intersecteze).

- Structura de dependență (T, P) definește un graf $G=(V, E)$, unde $V = \{1, 2, \dots, n, n+1\}$ și $E = \{(i, p_i) | 1 \leq i \leq n\}$. Acest graf trebuie să fie aciclic.

Spre exemplu, atunci când se ia în considerație mulțimea de etichete $T=\{NN, VBD, DT, IN\}$ și mulțimea de tipuri ale dependențelor $D=\{\text{sub, atr, det, pred}\}$, relațiile de dependență pentru propoziția

I	1	2	3	4	5	6	7
w_i	<i>the</i>	<i>Price</i>	<i>of</i>	<i>the</i>	<i>Stock</i>	<i>fell</i>	<i>EOS</i>

sunt descrise de structura (S, D) ; $S=(T, P)$, unde

$T = DT, NN, IN, DT, NN, VBD$

$P = 2, 6, 2, 5, 3, 7$

$D(1) = \text{det}, D(2) = \text{sub}, D(3) = \text{atr}, D(4) = \text{det}, D(5) = \text{det},$

$D(6) = \text{pred}$

sau, mai compact:

i	1	2	3	4	5	6	7
w_i	<i>the</i>	<i>Price</i>	<i>of</i>	<i>the</i>	<i>Stock</i>	<i>fell</i>	<i>EOS</i>
t_i	DT	NN	IN	DT	NN	VBD	EOS
p_i	2	6	2	5	3	7	0
d_i	det	Sub	atr	det	Det	pred	EOS

Se observă că relațiile de dependență astfel definite sunt în concordanță cu principiile generale enunțate anterior.

Având această definiție formală a relațiilor de dependență, putem spune că o **gramatică de dependență** este o structură (R, C) , unde R este o mulțime de restricții $R \subset T \times T \cup T \times W \cup W \times T \cup W \times W \cup D$, iar C este o mulțime de cerințe $C \subset T \times T \cup T \times W \cup W \times T \cup W \times W \cup D$.

O **structură sintactică** (S, D) ; $S = (T, P)$ relativ la o propoziție w_1, w_2, \dots, w_n este *corectă* din punctul de vedere al gramaticii (R, C) dacă

- $\forall 1 \leq i \leq n$,

$$(t_i, t_{p_i}, D(i)) \in R \vee (t_i, w_{p_i}, D(i)) \in R \vee (w_i, t_{p_i}, D(i)) \in R \vee (w_i, w_{p_i}, D(i)) \in R$$

(Restricțiile sunt îndeplinite.)

- $\forall 1 \leq i \leq n$

- dacă $\exists (t_i, t, d) \in C$, atunci $\exists 1 \leq j \leq n$ a.î. $p_j = i$, $t_j = t$, $D(j) = d$;

- dacă $\exists (t_i, w, d) \in C$, atunci $\exists 1 \leq j \leq n$ a.î. $p_j = i$, $w_j = w$, $D(j) = d$;

- dacă $\exists (w_i, t, d) \in C$, atunci $\exists 1 \leq j \leq n$ a.î. $p_j = i$, $t_j = t$, $D(j) = d$;

- dacă $\exists (w_i, w, d) \in C$, atunci $\exists 1 \leq j \leq n$ a.î. $p_j = i$, $w_j = w$, $D(j) = d$.

(Cerințele sunt îndeplinite.)

Restricțiile impun ca numai anumite relații dintre cuvinte să fie considerate valide, în timp ce cerințele permit anumitor cuvinte sau tipuri de cuvinte să impună existența în cadrul propoziției a altor cuvinte sau tipuri de cuvinte care trebuie să îndeplinească anumite relații.

În acest cadru se recomandă ca procesul de analiză sintactică să se desfășoare în două etape. Astfel, într-o primă etapă, ar trebui determinată structura de dependență (T, P) , după care, în etapa următoare, ar trebui stabilit tipul dependențelor D . După ce structura de dependență este cunoscută, pentru a stabili tipul relației de dependență corespunzătoare fiecărei perechi de cuvinte (w_i, w_{p_i}) , trebuie avută în vedere, între altele, funcția gramaticală, precum și faptul că aceasta este întotdeauna privită ca fiind funcția cuvântului dependent în relația sa cu capul.

Principiul conform căruia fiecare propoziție are o structură de dependență în cadrul căreia nici un arc (săgeată) nu se intersectează cu un altul și există câte o săgeată care țintește spre fiecare cuvânt este crucial în analiza sintactică de dependență și a fost numit de Richard Hudson "the No-tangling Principle"

(**principiul neîncălcării**). Acest principiu se bazează pe *tendința grupurilor sintactice de a fi continue*. Totuși, trebuie observat faptul că aceasta este numai o tendință, întrucât sunt cunoscute numeroase excepții. Acest principiu ar trebui, prin urmare, interpretat ca stipulând faptul că există o structură de tip schelet a propoziției, care ține laolaltă toate cuvintele acesteia fără apariția fenomenului de încălcire sau intersectare, alte dependențe putând fi însă adăugate acestei structuri. Cea mai importantă *excepție* la acest principiu este *coordonarea*, pe care R. Hudson propune să o tratăm prin recunoașterea unor șiruri sau secvențe de cuvinte, cum ar fi "Ion și Maria" din exemplul care urmează, alcătuite din alte șiruri mai mici, numite "conjuncți". Diagramele corespunzătoare ar putea fi de tipul

{{*Ion*} și {*Maria*}} *au venit*.

unde *conjuncții* sunt *Ion* și respectiv *Maria*. Un alt exemplu de coordonare este cel din fraza

Noi {{*am făcut un duș*} și {*am împachetat bagajele*}} *înainte de a pleca*.

O coordonare este în mod normal semnalată printr-o conjuncție, aceasta fiind plasată de obicei la începutul ultimului conjunct. Conjuncții pot fi șiruri de cuvinte, ca în cel de-al doilea exemplu, nu neapărat cuvinte individuale. Este evident faptul că fenomenul coordonării reprezintă o excepție la teoria relațiilor de dependență, întrucât dependența implică subordonare (a cuvântului dependent față de cap), cel puțin din punct de vedere sintactic dacă nu și semantic, în timp ce în cazul coordonării conjuncții au roluri sensibil egale. Tocmai de aceea, în aplicarea principiului neîncălcării, fiecare conjunct trebuie tratat în mod separat.

2. Relații de dependență în limba română

În stabilirea celor mai frecvente relații de dependență în limba română¹ s-a luat în considerație [1], [2], de cele mai multe ori, funcția sintactică a cuvântului

¹ Acest studiu a fost realizat în cadrul unui proiect de cercetare-dezvoltare finanțat de către Fundația Volkswagen pe o perioadă de doi ani (1996-1998). Adaptarea teoriei gramaticilor de dependență limbii române s-a făcut ca parte integrantă a conceperii sistemului german-bulgar-român de traducere asistată de calculator numit DBR-MAT. Scopul declarat al DBR-MAT a fost implementarea pilot a unui sistem de traducere asistată de calculator care să combine o abordare bazată pe procesarea cunoștințelor cu metode statistice în prelucrarea limbajului natural.

dependent. Astfel, au fost urmate o serie de reguli generale, aceasta fiind cel mai frecvent utilizată dintre ele.

Regulile generale de bază care au fost concepute și aplicate pentru stabilirea celor mai frecvente relații de dependență în limba română sunt următoarele:

- considerarea funcției sintactice (date de analiza sintactică clasică) a cuvântului dependent;
- în acele cazuri în care cuvântul dependent este o prepoziție sau o conjuncție coordonatoare, stabilirea tipului dependenței se face în concordanță cu funcția sintactică (clasică) a elementului (cuvântului) introdus în propoziție de către acea prepoziție sau conjuncție;
- caracteristicile morfologice ale cuvântului dependent sunt uneori luate în considerație (exemplu: *relația nehotărâtă*, stabilită atunci când cuvântul dependent este un articol nehotărât);
- tipul dependenței este dat de cuvântul cap numai în acele cazuri în care capul este reprezentat de o prepoziție sau o conjuncție coordonatoare (exemplu: *relația prepozițională*, *relația conjuncțională*);

Relațiile de dependență cel mai frecvent întâlnite în cazul limbii române și stabilite conform acestor reguli generale sunt prezentate în Tabelul 1. În mod evident, aceste dependențe ar putea fi în continuare rafinate în funcție de categoria gramaticală a cuvântului dependent (care poate fi substantiv, pronume, verb, adjectiv etc.), luând astfel naștere relații ca "relația prepozițional-adjectivală" (în care cuvântul cap este o prepoziție, iar cuvântul dependent este un adjectiv). O asemenea rafinare nu se recomandă însă atunci când nu este disponibil un corpus de dimensiuni semnificativ de mari.

Tabelul 1 include cele mai frecvente relații de dependență găsite, în cadrul de lucru oferit de către proiectul DBR-MAT, corespunzător limbii române. Menționăm faptul că, în acest cadru, au fost studiate numai texte științifice din domeniul chimiei și al tehnologiilor chimice. Tipurile de dependențe au fost stabilite conform regulilor generale menționate anterior, iar tabelul clasifică relațiile de dependență corespunzătoare în funcție de cuvântul cap. Precizăm, de asemenea, faptul că, în cadrul Tabelului 1, ori de câte ori se face referire la un cuvânt cap de tip verb, în afara cazului în care se specifică altceva, verbul respectiv poate fi atât predicativ, cât și nepredicativ. Tabelul 1 va fi extins și republicat pe web, fiind îmbogățit cu noi relații de dependență tipice limbii române, definite pe baza adnotării unor texte din ziare de largă circulație (a se vedea exemplele de texte românești adnotate).

TABELUL 1

CUVÂNT CAP	CUVÂNT DEPENDENT	TIPUL RELAȚIEI DE DEPENDENȚĂ	ABREVIERE
verb	prepoziție	complement circumstanțial	C.C.
verb	prepoziție	complement prepozițional	C.P.
verb	prepoziție	complement de agent	C.A.
verb	prepoziție	complement direct	C.D.
verb	verb (participiu)	complement direct	C.D.
verb (participiu)	verb auxiliar	relație auxiliară	AUX.
verb (infinitiv)	"a"	relație infinitivală	INF.
verb	conjunctie coordonatoare	complement direct	C.D.
verb	adverb	complement circumstanțial	C.C.
verb	pronume reflexiv	relație reflexivă	REF.
verb	substantiv	subiect	S.
verb	pronume	subiect	S.
verb	numeral	subiect	S.
verb	verb nepredicativ	subiect	S.
verb	substantiv	complement direct	C.D.
verb	pronume	complement direct	C.D.
verb	numeral	complement direct	C.D.
verb	verb nepredicativ	complement direct	C.D.
verb	substantiv	complement indirect	C.I.
verb	pronume	complement indirect	C.I.
verb	substantiv	nume predicativ	PRED.
verb	adjectiv	nume predicativ	PRED.
verb	adverb	nume predicativ	PRED.
verb	pronume	nume predicativ	PRED.

verb	numeral	nume predicativ	PRED.
verb	verb nepredicativ	nume predicativ	PRED.
prepoziție	substantiv	relație prepozițională	PREP.
prepoziție	conjunctie coordonatoare	relație prepozițională	PREP.
prepoziție	pronume demonstrativ	relație prepozițională	PREP.
prepoziție	verb (infinitiv)	relație prepozițională	PREP.
conjunctie coordonatoare	substantiv	relație conjuncțională	CONJ.
conjunctie coordonatoare	verb	relație conjuncțională	CONJ.
conjunctie coordonatoare	adjectiv participial	relație conjuncțională	CONJ.
adjectiv participial	prepoziție	complement de agent	C.A.
adjectiv participial	substantiv	complement indirect	C.I.
adjectiv participial	prepoziție	complement circumstanțial	C.C.
adjectiv participial	conjunctie coordonatoare	complement indirect	C.I.
adjectiv participial	adverb	complement circumstanțial	C.C.
adjectiv	prepoziție	complement prepozițional	C.P.
adjectiv	adverb	relație comparativă	COMP.
adjectiv	articol demonstrativ	relație comparativă	COMP.
substantiv	articol nehotărât	relație nehotărâtă	NEHOT.
substantiv	prepoziție	atribut substantival	A.S.
substantiv	conjunctie coordonatoare	atribut substantival	A.S.
substantiv	substantiv	atribut substantival apozitional	A.S.AP.
substantiv	substantiv	atribut substantival	A.S.

substantiv	adjectiv	atribut adjectival	A.A.
substantiv	pronume	atribut pronominal	A.P.
adverb	adverb	relație comparativă	COMP.
adverb	articol demonstrativ	relație comparativă	COMP.

3. Concluzii

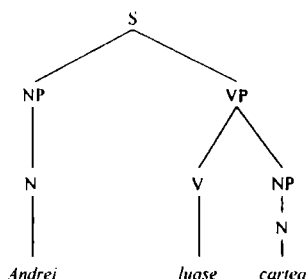
Așa cum se cunoaște deja, există două metode diametral opuse de a descrie structura sintactică a propozițiilor aparținând limbajului natural: utilizarea D-arborilor și respectiv a PS-arborilor. În mod evident, sunt posibile combinații ale celor două metode, care presupun anumite linii de compromis în diverse momente ale analizei, dar nu există nici o a treia posibilitate complet distinctă.

După cum se arată în [6], există câteva diferențe majore între **D-limbaj** (limbajul gramaticilor de dependență) și **PS-limbaj** (limbajul gramaticilor PS), pe care vom încerca să le menționăm, pe scurt, în continuare. Precizăm că, în cele ce urmează, arborele de derivare rezultat în urma efectuării analizei sintactice care utilizează o gramatică PS va fi denumit **PS-arbore**, în timp ce arborele corespunzător rezultat în urma utilizării în analiza sintactică a unei gramatici de dependență va fi numit **D-arbore**.

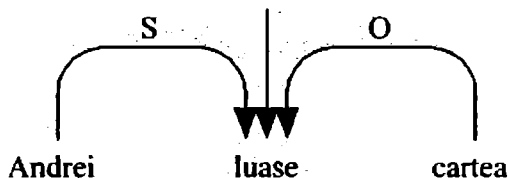
PS-structura propoziției

Andrei luase cartea

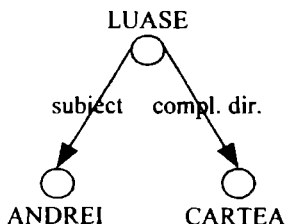
este reflectată de următorul **PS-arbore**



în timp ce **D-structura** aceleiași propoziții



este reflectată de următorul **D-arbore**:



O primă diferență semnificativă între D-limbaj și PS-limbaj constă în aceea că un PS-arbore corespunzător unei expresii aparținând limbajului natural arată care elemente ale acesteia (cuvinte sau chiar grupuri sintactice) se pot combina cu alte elemente pentru a forma niște unități de ordin mai mare. Un PS-arbore dezvăluie structura unei propoziții în termeni de grupări ale elementelor sale: blocuri maxime care constau din blocuri mai mici, care, la rândul lor, constau din blocuri și mai mici etc. PS-structura se exprimă în termeni de constituenți, operația logică aflată la baza acestei abordări fiind aceea a incluziunii de mulțimi, cu ajutorul căreia se exprimă apartenența la un grup sintactic, la o categorie etc.. Această abordare favorizează punctul de vedere *analitic*. Un D-arbore, pe de altă parte, arată ce elemente se află în relație cu alte elemente și în ce mod. D-structura propoziției reflectă relațiile existente între unități sintactice indivizibile, lucrând direct cu forme lexicale. În această abordare, operația logică de bază este aceea a stabilirii de relații binare. Propoziția nu mai este alcătuită din grupuri sintactice, categorii, ci din cuvinte legate între ele prin relații de dependență. Această abordare favorizează, prin urmare, punctul de vedere *sintetic*.

O altă diferență între PS-limbaj și D-limbaj este dată de faptul că, în cadrul unui PS-arbore, apartenența la o anumită categorie este specificată ca parte a reprezentării sintactice. Simboluri ca NP, VP, N etc. intervin în PS-arbori ca etichete ale unor vârfuri. Cu alte cuvinte, unele caracteristici sintactice date de operații precum categorizarea și subcategorizarea sunt folosite ca instrument principal în exprimarea rolului sintactic. În cadrul unui D-arbore, pe de altă parte, simbolurile reprezentând apartenența la o categorie, precum și alte proprietăți sintactice nu sunt admise ca elemente imediate ale structurii sintactice. (Astfel de informații sunt incluse în dicționar, lexicon etc.).

O a treia diferență esențială constă în faptul că, într-un PS-arbore, majoritatea nodurilor corespund unor simboluri neterminale. Ele reprezintă grupuri sintactice și nu corespund formelor lexicale efective care intervin în propoziția analizată. Prin contrast, un D-arbore conține numai noduri terminale, nefiind necesară nici o reprezentare abstractă a grupurilor sintactice.

PS-limbajul este, în esență, un limbaj linear, în timp ce D-limbajul este unul bidimensional, aceasta generând o altă deosebire fundamentală între cele două tipuri de reprezentări sintactice discutate aici. Astfel, în cadrul unui PS-arbore, vârfurile arborelui trebuie să fie ordonate linear, ordinea nefiind neapărat cea a formelor lexicale care intervin în propoziție. În cadrul unui D-arbore, pe de altă parte, vârfurile nu se află într-o astfel de ordine. Ordinea liniară a formelor lexicale din interiorul propoziției este un mijloc folosit de limbile naturale pentru a codifica relații sintactice și, prin urmare, ordinea liniară nu trebuie să intervină în structurile sintactice.

În fine, o ultimă deosebire importantă între cele două tipuri de reprezentări constă în aceea că, în timp ce un PS-arbore nu specifică tipul legăturii sintactice existente între doi constituenți, un D-arbore pune în mod special accentul pe specificarea în detaliu a tipului legăturii dintre oricare două elemente aflate în relație de dependență.

PRECIZĂRI:

Autorii doresc să mulțumească Prof. univ. dr. Theodor Hristea și Asist. univ. drd. Cristian Moroianu pentru consultațiile lingvistice acordate.

BIBLIOGRAFIE:

1. HRISTEA, F., On WG syntactic analysis with special reference to Romanian. *Analele Universității București, matematică-informatică*, 1, 1998, p. 59 - 69.
2. HRISTEA, F., POPESCU, M., A Word Grammar approach to syntactic analysis with special reference to Romanian. *Analele Universității București, matematică-informatică, Special Issue*, 1998, p. 101 - 113.
3. HUDSON, R., *Word Grammar*. Oxford, Blackwell, 1984.
4. HUDSON, R., *English Word Grammar*. Oxford, Blackwell, 1990.
5. HUDSON, R., *English Grammar*. London, Rontledge, 1998.
6. MEL'CUK, I. A., *Dependency Syntax: Theory and Practice*. Buffalo, Suny Press, 1987.

Dependency Grammar Annotator

Marius Popescu

Introducere

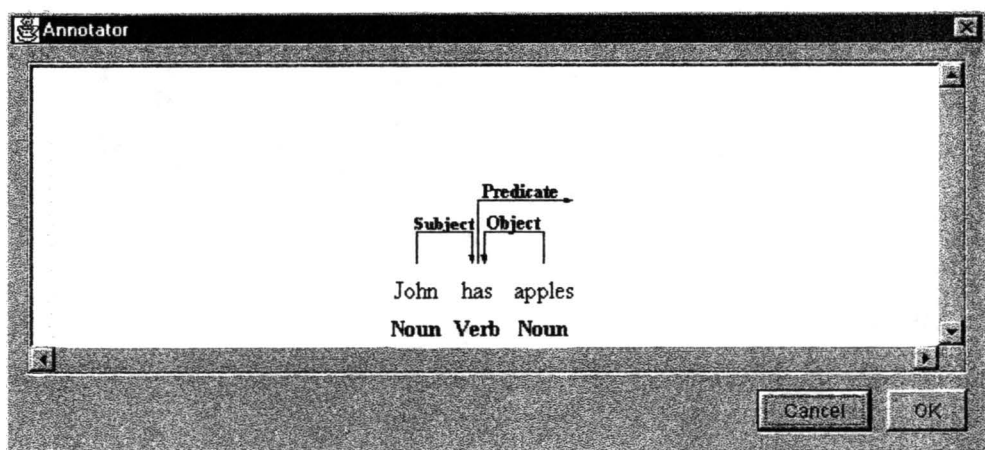
Ce este DGA

Dependency Grammar Annotator (**DGA**) este un instrument conceput pentru a facilita operatia de adnotare sintactica a textelor (a unui corpus) in cadrul formal al gramaticilor de dependenta.

Conform EAGLES¹: "*Adnotarea sintactica* este actiunea prin care se adauga informatii sintactice unui corpus, incorporand in text indicatori ai structurii sintactice cum ar fi: parantezari etichetate sau simboluri care sa indice relatiile de dependenta dintre cuvinte". Desi foarte folositoare in practica (testarea diverselor teorii gramaticale, achizitionarea automata de gramatici etc.) aceste corpusuri sunt costisitoare deoarece operatia de adnotare sintactica este mare consumatoare de timp si efort din partea celui / celor care adnoteaza. **DGA** a fost proiectat cu scopul de a minimiza efortul uman depus pe parcursul procesului de creare a unui corpus.

DGA este o interfata grafica usor de folosit care permite crearea si manipularea eficienta a structurilor sintactice. Deoarece formalismul in care se lucreaza este cel al gramaticilor de dependenta, aceste structuri sintactice constau in *relatiile de dependenta* formate din cuvintele unei propozitii etichetate cu partile de vorbire corespunzatoare si relatiile gramaticale care exista intre aceste cuvinte. In mod traditional relatiile de dependenta sunt indicate prin arce care leaga cuvantul dependent de cel pe care il determina, arcele fiind etichetate cu numele relatiei care exista intre cuvintele pe care le leaga. O astfel de reprezentare grafica (fiind conforma si cu recomandarile EAGLES) este folosita de **DGA** ca suport pentru operatia de adnotare.

¹ Expert Advisory Group on Language Engineering Standards
(<http://www.ilc.pi.cnr.it/EAGLES/home.html>)



Pe tot parcursul procesului de adnotare, utilizatorul operează direct asupra acestei reprezentări grafice. Datorită acestui lucru, în afară de comoditatea în utilizare, crește și acuratețea adnotării, deoarece utilizatorul are un feedback grafic imediat în ceea ce privește orice schimbare pe care o face în structura sintactică. Operarea asupra structurii sintactice este extrem de ușoară și intuitivă: pentru a crea o relație de dependență este nevoie doar de două clicuri de mouse (pe cele două cuvinte între care se dorește crearea relației), iar pentru etichetarea unui cuvânt cu o parte de vorbire sau pentru stabilirea tipului unei relații de dependență este nevoie doar de un clic și selectarea dintr-o listă a etichetei respective. Astfel, **DGA** permite o adnotare rapidă a textelor.

Considerăm că **DGA** răspunde la cerințele pe care Marcus și alții le-au identificat ca fiind importante în cadrul procesului de adnotare:

- **Acuratețea** - datorată faptului că se lucrează direct asupra reprezentării grafice și datorită feedbackului grafic imediat pe care **DGA** îl oferă utilizatorului.
- **Viteza** - crearea și manipularea relațiilor de dependență se face extrem de rapid cu ajutorul mouse-ului.
- **Consistența** - utilizatorul își stabilește setul de parti de vorbire și relații de dependență după care, pentru a le folosi, nu trebuie decât să le selecteze din diferite liste.

Caracteristici

- **Usurinta în folosire:** faptul că se operează direct asupra reprezentării grafice implică o mare ușurință în folosire și o viteză de lucru sporită (pentru mai multe amănunte vezi Ce este DGA).

- **Portabilitate:** **DGA** a fost scris in Java 2. Fiind o aplicatie Java pura, **DGA** poate rula practic pe orice platforma / sistem de operare pentru care exista mediu de executie Java (JRE) 2 (a fost testat pe sistemele Windows 95/98/NT si Linux). Deoarece foloseste tehnologia *pluggable look and feel* **DGA** se va comporta din punctul de vedere al interfetei ca o aplicatie nativa pe platforma pe care ruleaza, astfel utilizatorul fiind deja obisnuit cu elementele de baza ale interfetei: meniuri, butoane, casete de dialog standard etc.
- **Conformitate cu standardele actuale:** **DGA** respecta recomandarile EAGLES referitoare la adnotarea sintactica. Textele adnotate sunt salvate in format XML, standardul in descrierea datelor adoptat si de comunitatea lingvistica ca modalitate standard de reprezentare a corpusurilor. Desi pentru adnotarea sintactica nu exista inca un set standard de taguri XML, asa cum exista pentru adnotarea morfosintactica XCES², **DGA** foloseste un set minimal de taguri inspirat din XCES. Astfel, fisierele XML produse de **DGA** pot fi transformate usor cu ajutorul XSLT in fisiere XML bazate pe alt vocabular (set de taguri) care sa raspunda nevoilor utilizatorului sau sa fie conforme cu un standard viitor. Mai multe amanunte tehnice se gasesc in Formatul XML folosit de **DGA**.
- **Flexibilitate:** in afara de faptul ca analiza sintactica trebuie sa fie sub forma relatiilor de dependenta, **DGA** nu impune nici o alta restrictie utilizatorului. Acesta isi poate defini cu usurinta si modifica oricand propriul set de parti de vorbire si relatii de dependenta pe care le va folosi in adnotare.

Ghid de folosire

Instalare

Cerinte:

1. Pentru a putea rula, **DGA** are nevoie de mediul de executie Java 2. Deci sistemul de operare al calculatorului pe care se doreste instalarea lui **DGA** trebuie sa fie unul pentru care exista o implementare a lui Java 2 (Windows 95/98/ME/NT/000, majoritatea versiunilor de Unix, MacOS X).
2. Sistemul pe care se va instala **DGA** trebuie sa fie unul suficient de puternic (viteza procesor, memorie). In cazul unui PC sunt necesare minimum 133 MHz, 32M RAM.

² XML Corpus Encoding Standard (<http://www.cs.vassar.edu/XCES>)

Instalare:

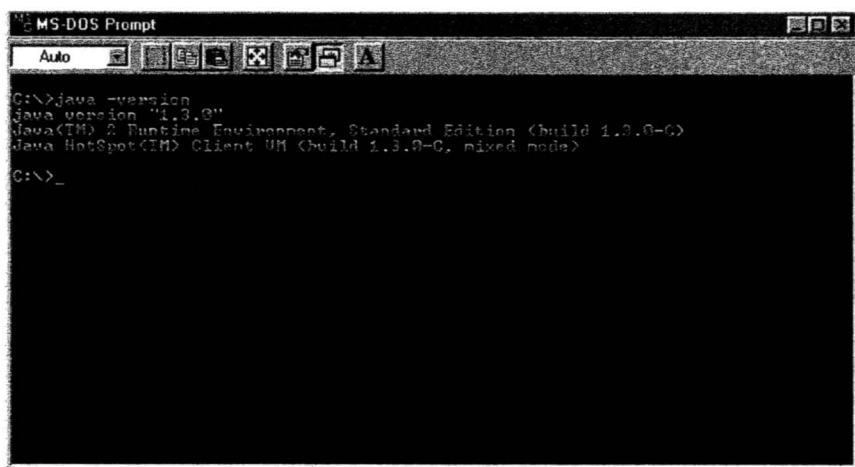
1. Instalati Java 2 pe sistemul dumneavoastra. Daca Java 2 este deja instalat, sariti peste acest pas. Pentru platformele Windows, Linux, Solaris, kiturile de instalare corespunzatoare se pot gasi la java.sun.com. Se poate instala intreg mediul de dezvoltare JDK sau doar mediul de executie JRE. Se recomanda folosirea versiunii 1.3 sau a uneia mai noi.
2. Asigurati-va ca este pusa in PATH calea catre executabilul java. In Windows 95/98/NT aceasta se poate face punand in `autoexec.bat` o linie de forma:

```
SET PATH=c:\path; %PATH%
```

unde `c:\cale` se inlocuieste cu calea actuala catre directorul unde se afla executabilul `java`. Dupa aceasta operatie (si restartarea sistemului), ar trebui ca la comanda (indiferent de directorul din care este data aceasta):

```
java -version
```

raspunsul sistemului sa fie asemanator cu:



```
MS-DOS Prompt
Auto
C:\>java -version
java version "1.3.0"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.0-GC)
Java HotSpot(TM) Client VM (build 1.3.0-GC, mixed mode)
C:\>_
```

3. Desfaceti arhiva `dga.zip` si plasati continutul acesteia oriunde doriti in structura de directoare.
4. Pentru a lansa **DGA** din command prompt (MS-DOS prompt) aflandu-va in directorul `DGAnnotor` (pentru a face ca directorul curent sa fie `DGAnnotor` executati `cd cale\catre\DGAnnotator`), lansati comanda:

```
java -classpath  
dga.jar;crimson.jar;xalan.jar;jaxp.jar DGAnnotator
```

Alternativ puteti lansa in executie fisierul **dga.bat** din acelasi director. Pentru a face lucrul pe viitor mai comod puteti crea un shortcut la fisierul **dga.bat** (pe care sa-l puneti pe desktop), la proprietatile shortcutului setati ca director de lucru directorul DGAnnotator, iar ca icon fisierul **dga.ico** din acest director. Din acest moment aplicatia **DGA** poate fi lansata cu un dublu clic pe

icon-ul

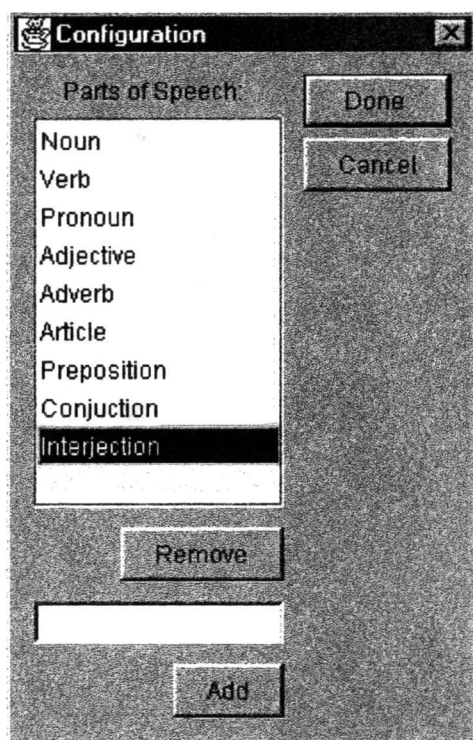


Nota: Instructiunile de mai sus s-au concentrat mai mult pe instalarea sub sistemul de operare Windows. Pentru a instala si rula **DGA** pe orice sistem, in esenta, trebuie sa instalati mediul Java 2 pe acel sistem, sa desfaceti arhiva dga.zip si apoi sa fiti capabili sa rulati clasa (Java) DGAnnotator.

Configurare

Dupa instalare **DGA** pune la dispozitia utilizatorului un set initial de parti de vorbire si relatii de dependenta. Acest set are doar un rol ilustrativ si nu are nici o justificare lingvistica. Utilizatorul trebuie sa-si defineasca propriul set de parti de vorbire si relatii de dependenta. Aceasta operatie de configurare (stergere sau adaugare de parti de vorbire sau relatii de dependenta) poate fi facuta oricand, nu neaparat la inceputul procesului de adnotare. Evident, probabil ca se va porni cu un prim set de parti de vorbire si relatii de dependenta, iar pe parcurs, daca se constata ca mai este nevoie de o noua parte de vorbire sau relatie de dependenta, aceasta va putea fi adaugata atunci. Or dimpotriva, o parte de vorbire sau relatie de dependenta poate fi stearsa oricand se constata ca respectiva parte de vorbire sau relatie de dependenta nu a fost niciodata folosita si nici nu se crede ca va fi folosita in viitor.

Pentru a configura (sterge / adauga) setul de parti de vorbire, din meniul **Configuration**, se selecteaza comanda **Parts of Speech**. Va apare o caseta de dialog ca cea de mai jos:




Pentru a sterge o parte de vorbire, aceasta trebuie selectata din lista (cu un clic de mouse pe respectiva parte de vorbire), dupa care se apasa butonul **Remove**. Immediat partea de vorbire respectiva va disparea din lista. Pentru a adauga o parte de vorbire, se tasteaza numele acesteia in campul de deasupra butonului **Add** si se apasa butonul **Add**. Immediat, aceasta parte de vorbire va aparea la sfarsitul listei. Aceste doua operatii se pot repeta de oricate ori (in orice ordine) se doreste. Cand lista ajunge sa fie cea dorita se poate apasa butonul **Done**; caseta de dialog va disparea, iar noua lista a partilor de vorbire devine imediat disponibila pentru adnotare. Evident, se poate renunta in orice moment la operatia de configurare, apasand butonul **Cancel**.

Pentru a configura (sterge / adauga) setul de relatii de dependenta, din meniul **Configuration**, se selecteaza comanda **Dependency Relations**. Va aparea o caseta de dialog identica cu cea de mai sus, in care lista partilor de vorbire va fi inlocuita cu lista relatiilor de dependenta. Mai departe, modul de operare este identic cu cel pentru stergerea si adaugarea partilor de vorbire.


Deschiderea unui document pentru adnotare

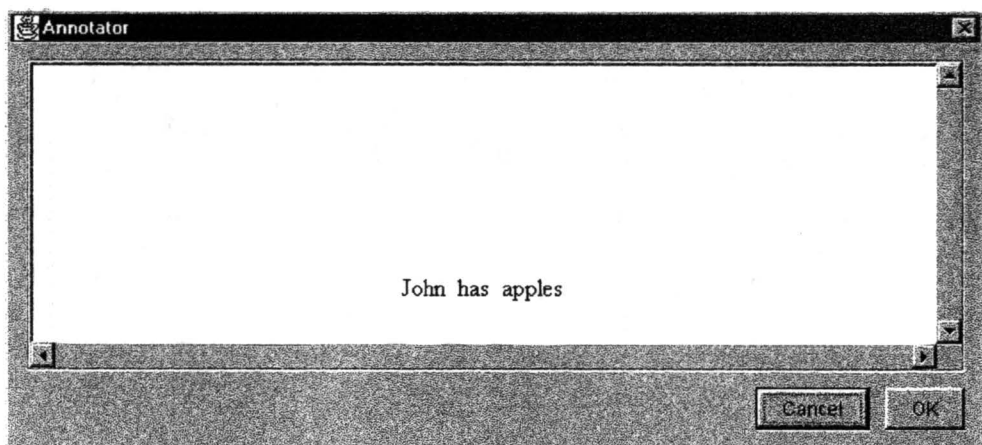
Pentru a adnota un text, trebuie mai intai deschis fisierul care contine textul (fisierul trebuie sa fie de tip text). Aceasta operatie se face alegand comanda **Open**

text din meniul **File** sau apasand butonul  din bara de instrumente. Se va deschide o caseta de dialog standard (pentru platforma respectiva), care permite selectarea unui fisier pentru deschidere. Continutul fisierului ales va fi afisat intr-o fereastră, de unde utilizatorul poate selecta propozitii pentru a le adnota.

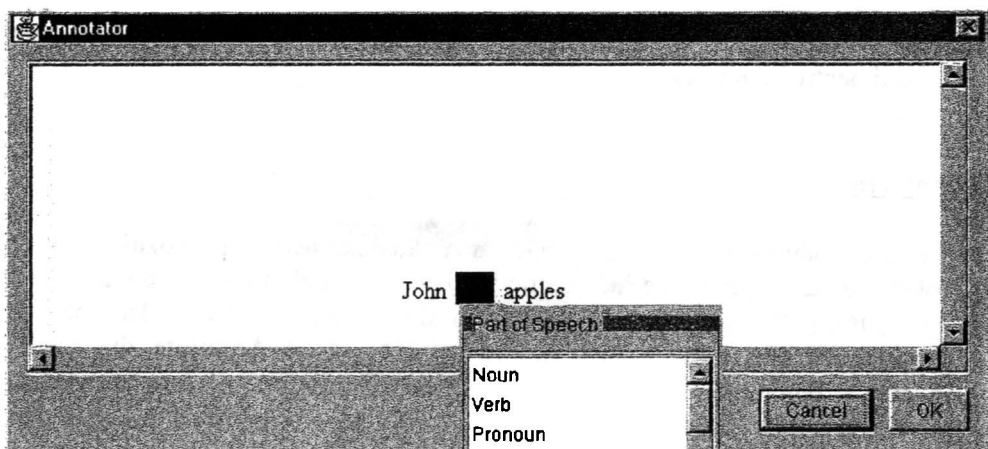
Nota: Utilizatorul nu poate modifica continutul fisierului text afisat, singura operatie permisa fiind cea de selectie a unei portiuni de text (selectarea unei propozitii pentru adnotare).

Adnotarea

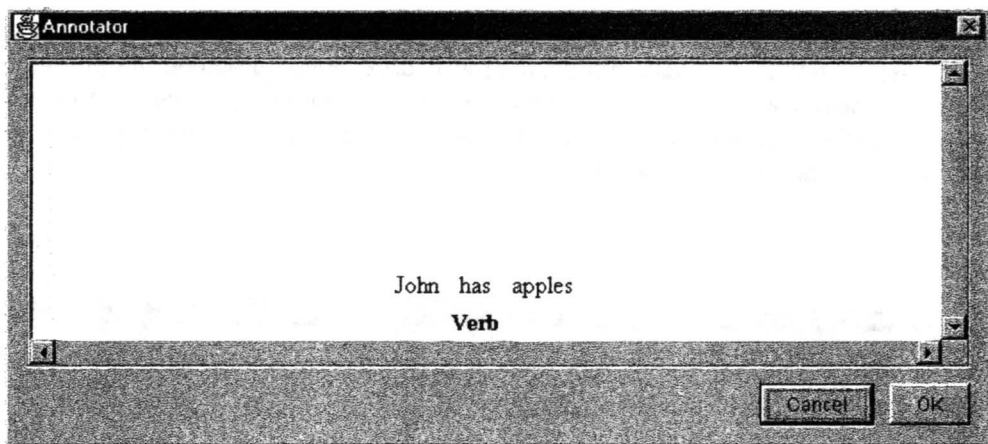
Procesul de adnotare incepe prin selectarea din text a unei propozitii ce va fi adnotata. Aceasta operatie se face in mod obisnuit tragand mouse-ul cu butonul din stanga apasat peste propozitia (textul) care se doreste a fi selectata. Odata selectata o portiune de text (propozitie), se activeaza comanda **Annotate** din meniul **Annotate** si butonul  din bara de instrumente. Apasand acest buton sau alegand comanda **Annotate** din meniul **Annotate** se va deschide o caseta de dialog in care, intr-un camp de editare, este continut textul care a fost selectat. Aici utilizatorul poate edita textul. Desi utilizatorul poate sterge complet textul din campul de editare si scrie altul nou, nu acesta a fost scopul pentru care a fost prevazuta aceasta caseta de dialog. Caseta de dialog a fost prevazuta pentru a da posibilitatea utilizatorului sa faca mici modificari asupra propozitiei ce va fi adnotata. Sa poata elimina semnele de punctuatie sau sa le separe de cuvinte, daca se doreste si adnotarea acestora etc. Odata terminate modificarile, apasand butonul **Annotate** va incepe procesul propriu-zis de adnotare. Textul selectat (si eventual modificat in caseta de dialog anterioara) va fi afisat intr-o noua fereastră. In aceasta fereastră se vor efectua toate operatiile de adnotare.



Pentru a stabili partea de vorbire a unui cuvânt, executați clic dreapta pe cuvântul respectiv (plasați mouse-ul pe cuvântul respectiv și apăsați butonul din dreapta al mouse-ului). Cuvântul pe care s-a făcut clic dreapta va fi marcat cu roșu și sub el se va deschide un meniu contextual care conține lista partilor de vorbire (cele stabilite de utilizator, vezi configurarea).

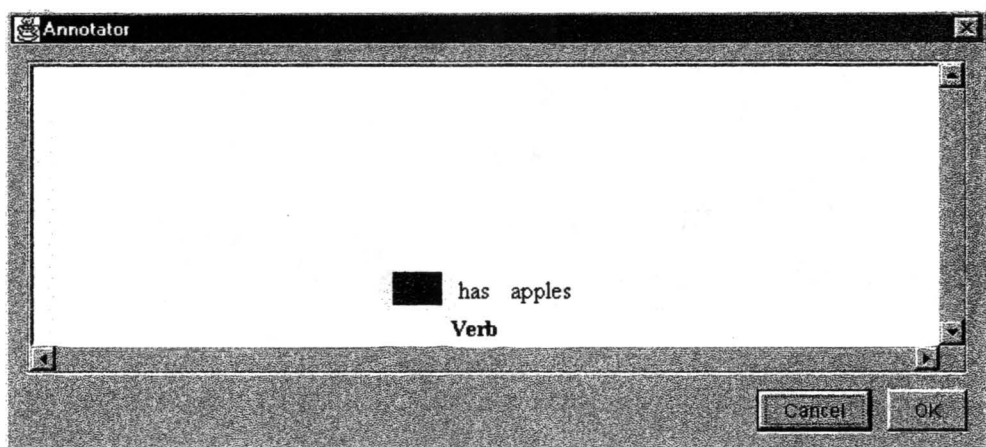


Selectați (clic pe) partea de vorbire dorită și aceasta va apărea imediat sub cuvântul pentru care s-a făcut operația.

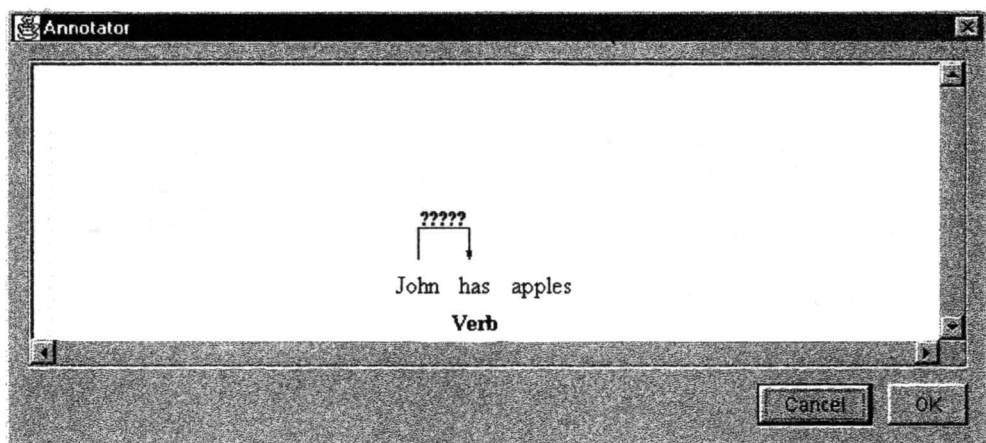


Dacă se dorește modificarea sau ștergerea părții de vorbire deja asociate unui cuvânt, un clic dreapta pe cuvântul respectiv sau pe partea de vorbire asociată acestuia, va deschide din nou meniul contextual cu lista partilor de vorbire. Selectând altă parte de vorbire, aceasta va înlocui imediat vechea parte de vorbire. Dacă din meniul contextual se apasă pe butonul **Delete Part of Speech** vechea parte de vorbire va fi ștearsă, cuvântul respectiv nu va mai avea asociată nici o parte de vorbire.

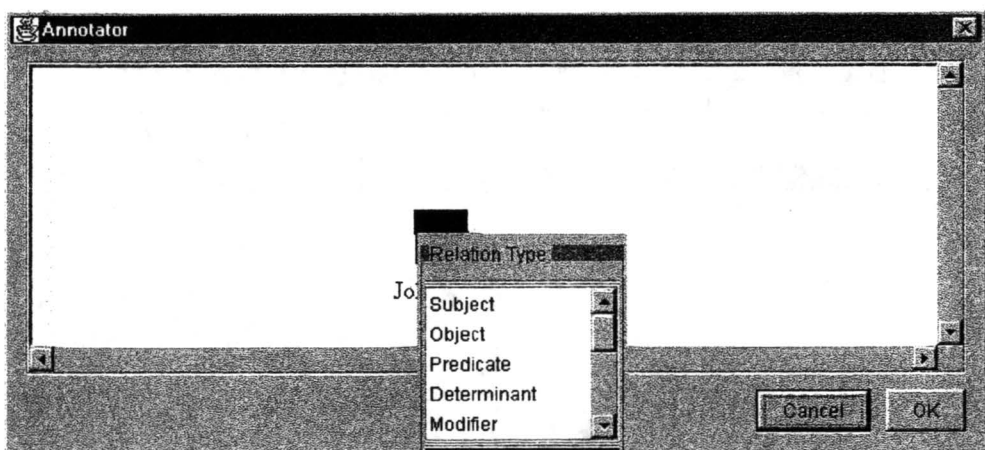
Pentru a crea o relatie de dependenta intre doua cuvinte se va face mai intai clic (buton stanga) pe cuvantul *dependent* (cel care determina, cel din care porneste arcul). Acest cuvint va fi marcat cu albastru.



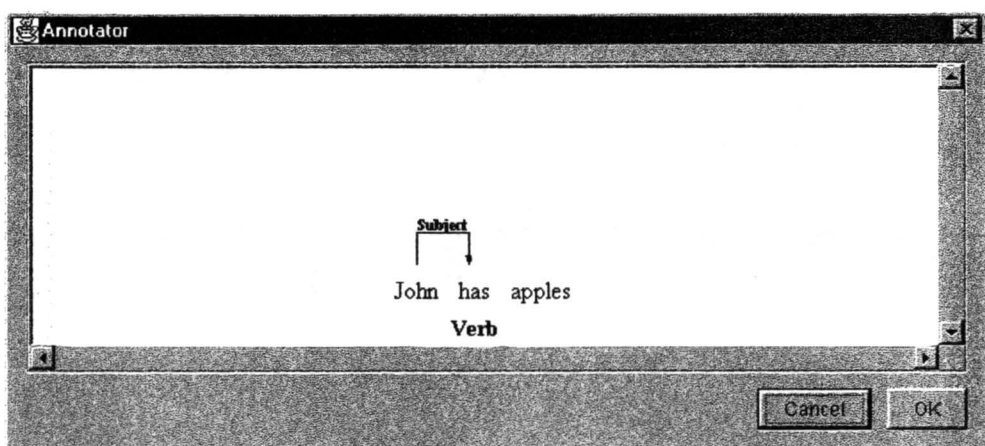
Apoi se va executa clic pe cuvantul *cap* (cel determinat, cel in care ajunge arcul). Imediat se va crea un arc de la cuvantul dependent la cuvantul cap, etichetat cu sirul "?????". Acest sir semnifica faptul ca relatiei de dependenta nu i-a fost inca stabilit tipul.



Pentru a stabili tipul unei relatii de dependenta, trebuie executat clic dreapta pe sirul "?????" de deasupra arcului ce reprezinta relatia. Se va deschide un meniu contextual de unde se poate alege tipul relatiei (tipuri stabilite de utilizator, vezi configurarea).

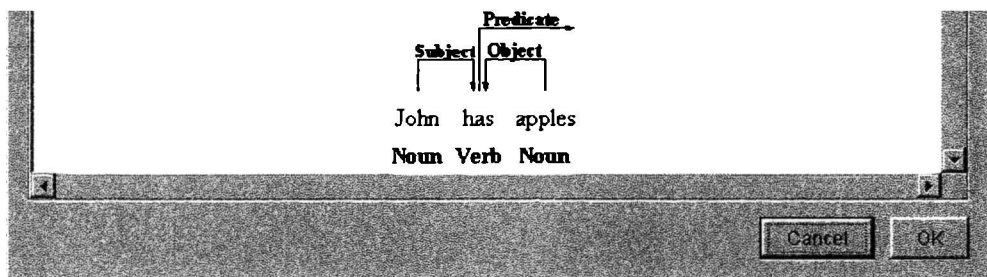


Selectand din lista tipul relatiei (ca si la stabilirea partii de vorbire), acesta va aparea imediat deasupra arcului ce reprezinta relatia careia i s-a stabilit tipul.



Ca si in cazul partilor de vorbire, daca se doreste modificarea tipului unei relatii de dependenta, trebuie executat clic dreapta pe tipul relatiei respective si meniul contextual cu lista tipurilor de relatii se va deschide din nou. Selectand un alt tip de relatie, acesta va inlocui imediat vechiul tip. Daca se va apasa butonul **Delete Relation** din acest meniu contextual, se va sterge intreaga relatie de dependenta (adica arcul).

Conform formalismului gramaticilor de dependenta, fiecare cuvnt dintr-o propozitie trebuie sa depinda de (sa determine un) altul. Exceptie face un singur cuvnt considerat principal, *capul* intregii propozitii (de obicei verbul principal), care nu depinde de nimeni. Acest lucru se marcheaza grafic printr-un arc ca cel atasat verbului **has** in figura de mai jos



Specificarea faptului ca un cuvânt este capul unei propozitii se face astfel: se face clic pe cuvântul respectiv si acesta este marcat cu albastru, apoi se face inca o data clic pe el si acesta va fi imediat semnalat ca fiind capul propozitiei printr-un arc, ca in figura de mai sus. Stabilirea tipului pentru acest arc se face ca si pentru celelalte relatii. **Atentie!** Pentru a marca un cuvânt ca fiind capul propozitiei nu faceti dublu clic pe el, ci clic, asteptati sa fie marcat cu albastru, si faceti inca o data clic. Deoarece o propozitie nu are decat un singur cuvânt cap, **DGA** nu permite definirea decat a unui singur cuvânt cap. Odata definit un cuvânt cap, succesiunea de operatii prin care se incearca definirea si a altui cuvânt ca fiind capul propozitiei nu va avea nici un efect.

Operatiile de stabilire a partii de vorbire a unui cuvânt, de creare a unei relatii de dependenta si de stabilire a tipului unei relatii de dependenta, se pot repeta in orice ordine se doreste, pana cand utilizatorul considera ca propozitia este adnotata. In acest moment, prin apasarea butonului **OK**, procesul de adnotare al respectivei propozitii se incheie. Evident, se poate renunta in orice moment la adnotarea respectivei propozitii, apasand **Cancel**. Daca se apasa **OK** si propozitia nu este adnotata complet (in sensul ca nu i s-a atribuit fiecarui cuvânt o parte de vorbire sau nu s-a stabilit pentru fiecare cuvânt de cine depinde si care este tipul respectivei relatii de dependenta), utilizatorul este avertizat, dar are posibilitatea sa pastreze propozitia incomplet adnotata.

Nota: De fiecare data cand a fost vorba de clic dreapta, ne-am referit de fapt la operatia de obtinere a unui meniu contextual care este clic dreapta pentru Windows si Unix (X Windows), dar care poate fi o alta in cazul altor platforme (de exemplu MacOS).

Salvarea datelor si iesirea

Propozitiile adnotate sunt pastrate in memorie. Ele nu sunt salvate intr-un fisier decat la apelarea explicita a comenzii **Save annotation** din meniul **File**. La apelarea acestei comenzi, se deschide o caseta de dialog standard (pentru platforma respectiva), unde utilizatorul, dupa ce selecteaza directorul unde se doreste plasarea fisierului cu propozitiile adnotate, trebuie sa scrie numele fisierului in care se doreste salvarea datelor si sa apese butonul **Save**. Fisierul trebuie sa aiba extensia **.xml** (acesta fiind formatul in care se salveaza datele). Utilizatorul poate, de asemenea, sa selecteze in caseta de dialog care se deschide un fisier deja existent, caz in care datele sunt adaugate la sfarsitul fisierului selectat (la cele existente deja in fisierul selectat). Este sarcina utilizatorului sa se asigure ca fisierul selectat contine tot propozitii adnotate si nu alt tip de date.

Fisierele xml in care se salveaza datele pot fi puse oriunde se doreste in structura de directoare, insa, pentru a putea fi utile, ele au nevoie de fisierul **dga.dtd**, care se gaseste in directorul aplicatiei **DGAnnotator**. Asa ca este bine ca acolo unde sunt plasate fisierele xml cu datele sa fie copiat si fisierul **dga.dtd**.

Terminarea unei sesiuni de lucru se face selectand comanda **Exit** din meniul **File** sau inchizand fereastra principala a aplicatiei. Daca la inchidere mai exista date (propozitii adnotate) care nu au fost inca salvate, utilizatorul este avertizat si are posibilitatea sa le salveze.

Sistemul de bookmark

In procesul de alcatuire a unui corpus de obicei se adnoteaza o cantitate mare de texte. Deoarece s-ar putea ca un text sa nu fie complet adnotat pe parcursul unei singure sesiuni de lucru, **DGA** este prevazut cu un sistem de *bookmark* care ii permite utilizatorului sa retina si sa revina la un anumit punct dintr-un anumit text, oricand doreste (pentru a continua adnotarea din acel punct).

Pentru a marca un anumit punct dintr-un text (cand textul este deschis pentru adnotare), se selecteaza o portiune de text (acesta va fi punctul de revenire) si se alege comanda **Add Bookmark** din meniul **Bookmarks**. In lista de bookmark-uri din meniul **Bookmarks** va aparea un bookmark nou, etichetat cu data si ora la care s-a facut bookmark-ul.

Pentru ca utilizatorul sa ajunga in punctul (text si pozitie in text) spre care indica un anumit bookmark, el trebuie sa selecteze respectivul bookmark din meniul **Bookmarks**. Daca bookmarkul ales indica un punct chiar din fisierul curent (cel care era deschis in acel moment), atunci textul va fi astfel positionat in fereastra (prin derulare), incat portiunea marcata (cea spre care indica bookmarkul) sa fie vizibila. Daca bookmarkul ales indica un punct dintr-un alt fisier decat fisierul

curent, atunci fisierul (textul) curent va fi inchis si se va deschide fisierul (textul) spre care indica bookmarkul, textul fiind pozitionat in fereastra astfel incat portiunea marcata (cea spre care indica bookmarkul) sa fie vizibila. Un bookmark poate fi urmat si daca nu este deschis nici un fisier (text), in acest caz deschizandu-se direct fisierul spre care indica bookmarkul.

Pentru a nu incarca exagerat meniul **Bookmarks** (si pentru ca in mod obisnuit nu este nevoie de foarte multe bookmarkuri), **DGA** limiteaza numarul maxim de bookmarkuri la 10. Utilizatorul are posibilitatea de a sterge bookmarkurile pe care nu le mai foloseste, alegand comanda **Remove Bookmark** din meniul **Bookmarks**. Se va deschide o caseta de dialog ce va contine lista bookmarkurilor existente. Selectand un bookmark din aceasta lista si apasand butonul **Remove** acesta va fi sters.

Vizualizarea si modificarea unui text adnotat

DGA permite vizualizarea si eventual modificarea (modificarea adnotarilor) textelor adnotate anterior. Pentru aceasta trebuie aleasa comanda **Open corpus** din meniul **File**. Se va deschide o caseta de dialog standard, care permite selectarea fisierului ce contine textul adnotat (corpusul). Fisierul ales trebuie sa fie un fisier xml, cu o structura conforma cu formatul **DGA** (vezi Formatul XML folosit de **DGA**).

Continutul fisierului ales va fi afisat intr-o fereastra astfel: textul va fi afisat ca o multime de propozitii, fiecare propozitie aparand ca un hyperlink (intr-un navigator web). Facand clic pe o propozitie, adnotarea acesteia va fi afistata, in forma grafica obisnuita, intr-o caseta de dialog. In aceasta fereastra pot fi facute si modificari asupra adnotarii, aici fiind permise toate operatiile de adnotare (vezi sectiunea referitoare la adnotare).

Salvarea modificarilor se face selectand comenzile **Save corpus** sau **Save corpus as** din meniul **File**, iar inchiderea fisierului se face cu comanda **Close corpus** din acelasi meniu.

Formatul XML folosit de DGA

Textele adnotate sunt salvate in format XML, standardul in descrierea datelor adoptat si de comunitatea lingvistica ca modalitate standard de reprezentare a corpusurilor. Desi pentru adnotarea sintactica nu exista inca un set standard de taguri XML, asa cum exista pentru adnotarea morfosintactica XCES, **DGA** foloseste un set minimal de taguri inspirat din XCES. Astfel, fisierele XML produse de **DGA** pot fi transformate usor cu ajutorul XSLT in fisiere XML bazate pe alt vocabular (set de taguri), care sa raspunda nevoilor utilizatorului sau sa fie conforme cu un standard viitor.

Pentru a ilustra setul de taguri folosit dam mai jos un fragment de fisier xml care reprezinta adnotarea propozitiei "John has apples" (vezi Ce este DGA).

```
<s>
  <tok>
    <orth>John</orth>
    <ordno>1</ordno>
    <ctag>Noun</ctag>
    <syn>
      <head>2</head>
      <reltype>Subject</reltype>
    </syn>
  </tok>
  <tok>
    <orth>has</orth>
    <ordno>2</ordno>
    <ctag>Verb</ctag>
    <syn>
      <head>4</head>
      <reltype>Predicate</reltype>
    </syn>
  </tok>
  <tok>
    <orth>apples</orth>
    <ordno>3</ordno>
    <ctag>Noun</ctag>
    <syn>
      <head>2</head>
      <reltype>Object</reltype>
    </syn>
  </tok>
</s>
```

Fiecare propozitie este marcata de tagul <s> ... </s>. Fiecare cuvant din propozitie impreuna cu informatiile referitoare la adnotarea acestui cuvant este marcat de tagul <tok> ... </tok>. In interiorul acestui tag, forma ortografica, asa cum apare in textul care este adnotat, este marcata cu tagul <orth> ... </orth>, tagul <ordno> ... </ordno> indica numarul de ordine al cuvantului in cadrul propozitiei (al catalea este de la inceputul propozitiei), cu tagul <ctag> ... </ctag> se specifica partea de vorbire, iar <syn> ... </syn> delimiteaza informatiile sintactice. In interiorul tagului <syn> ... </syn> se specifica cuvantul cap prin numarul sau de ordine in cadrul propozitiei, numar delimitat de tagul <head> ... </head>. Cu ajutorul tagului <reltype> ... </reltype> se specifica tipul relatiei de dependenta care exista intre cele doua cuvinte (cuvantul caruia ii apartine adnotarea si cuvantul cap).

Bibliografie

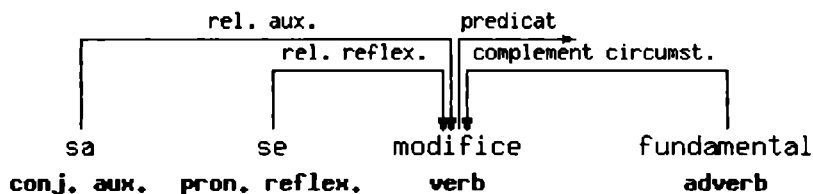
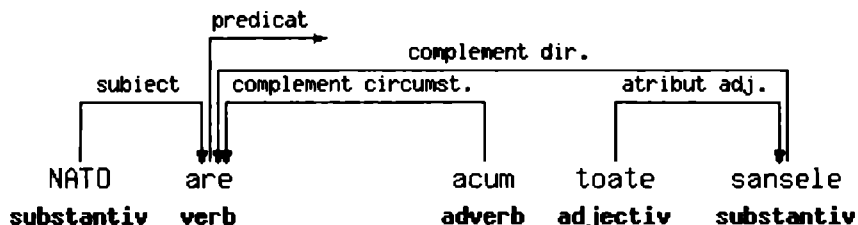
Hudson R. *English Word Grammar*. Oxford: Blackwell, 1990

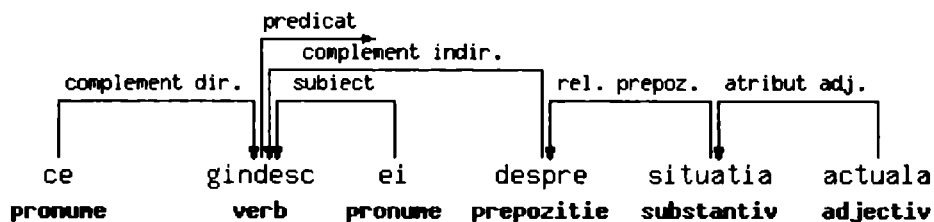
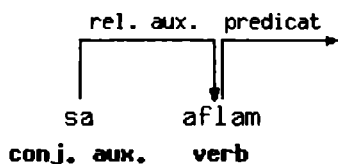
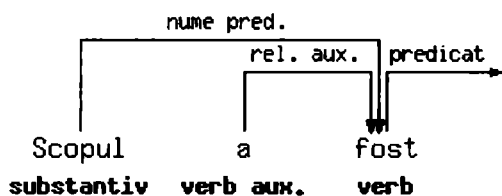
Marcus P.,Satornini B., Marcinkiewicz M. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313-330, 1993.

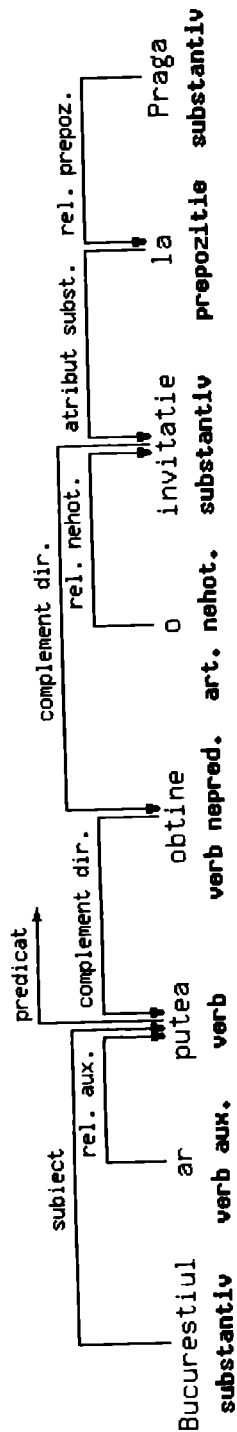
Tesniere L. *Elements de syntaxe structurale* Paris: Klincksieck, 1959

Anexa

Exemple de propozitii romanesti adnotate cu DGA







TEORII GRAMATICALE CONTEMPORANE: GRAMATICA CENTRILOR DE SINTAGMA (HPSG)

Ana-Maria Barbu și Emil Ionescu

1. INTRODUCERE

Anii '80 sunt anii unei veritabile explozii în domeniul teoriilor gramaticale. Teoriile apărute în această perioadă au două elemente comune: ele provin din aceeași matrice --lingvistica generativă-- și sunt versiuni ale aceluiași proiect care caracterizează în cel mai înalt grad mișcarea de idei din domeniul lingvisticii teoretice contemporane. Este vorba despre proiectul gramaticii universale.

Teoriile care s-au impus deja în peisajul actual al lingvisticii --pentru intervale de timp diferite și cu forțe inegale de penetrație-- sunt următoarele: Teoria Guvernării și a Dependențelor Anaforice (reformulată mai târziu ca "Teoria Principiilor și a Parametrilor" și devenită, în zilele noastre, "Programul Minimalist" --Chomsky 1981, 1995); Gramatica Lexico-Funcțională (Bresnan 1982), Gramatica Sintagmatică Generalizată (Gazdar, Klein, Pullum și Sag 1985), Gramatica Categoricală (Oehrle, Bach și Wheeler 1988) și Gramatica Centrilor de Sintagmă (Pollard și Sag 1987 , 1994). Ne propunem ca în rândurile următoare să facem o descriere a acestui ultim curent de cercetare.

Denumirea de "Gramatica Centrilor de Sintagmă" (abreviat HPSG) este o

traducere aproximativă a numelui original *Head-driven Phrase Structure Grammar*. Trăsăturile generale ale acestei teorii sunt următoarele:

(I) teoria se declară neutră față de problema dacă principiile și instrumentele pe care le întrebuințează sunt relevante pentru structurile mentale --preformate sau dobândite-- necesare întrebuințării limbajului;

(II) ea face parte din familia mai largă a acelor gramatici bazate pe conceptul de unificare (pentru acest concept, a se vedea în special Shieber 1986);

(III) teoria folosește un singur nivel de analiză (care într-o terminologie oarecum vagă este nivelul de suprafață);

(IV) ea este mai mult decât o sintaxă, deoarece prin modul ei de organizare încorporează și analiza semantică, dezvoltată în termenii teoriei semantice a situațiilor (Barwise și Perry 1983).

2. STRUCTURA HPSG

HPSG are următoarea structură generală:

- (i) un ansamblu de reprezentări care sunt clasificate;
- (ii) o "arhitectonică".

Arhitectonica înseamnă o structură de componente. Componentele sunt lexiconul, sintaxa și componenta aserțiunilor de precedență liniară.

Lexiconul cuprinde la rândul său reprezentări (intrări) și reguli lexicale. Sintaxa e alcătuită din reprezentări (reguli) sintagmatice și din principii formulate în legătură cu regulile asumate.

Din 1987 (data 'oficială' de apariție a teoriei), HPSG a suferit câteva reformulări locale, ce nu afectează caracteristicile enumerate în (I)-(IV). Am

optat pentru prezentarea variantei celei mai recente, care se găsește în capitolul al nouălea din Pollard și Sag 1994.

Prezentarea de față se dovedește de multe ori infidelă în raport cu textele de bază. Infidelitatea e de două feluri: pe de-o parte, în măsura posibilului, detaliile tehnice sunt evitate și se recurge la simplificări intuitive; pe de altă parte, expunerea atinge elementele esențiale ale teoriei, dar nu e completă. Cititorul nu va găsi, de pildă, aproape nimic despre interpretarea semantică, după cum nu va fi în posesia tuturor principiilor asumate în Pollard și Sag 1994.

Se obișnuiește ca descrierile care au ca obiect HPSG să se concentreze asupra asemănărilor și deosebirilor dintre această teorie și teoriile contemporane concurente (a se vedea, de pildă, Abeillé 1993). Nici în această privință nu vom urma tradiția. Vom căuta să luminăm punctele de intersecție dintre modul de a gândi faptele gramaticale propus de HPSG, și modul tradițional de abordare. Sperăm ca, prin aceasta, două dintre cele mai rezistente prejudecăți să fie ajutate a tinde către un relativism fructuos. Avem, desigur, în vedere ideea larg răspândită că nici o teorie de dată recentă nu mai poate aduce perspective care să nu fie deja conținute în modul tradițional de concepere a faptului gramatical, precum și opinia opusă că singurul adevăr științific se găsește în teritoriul contemporaneității. Sperăm, totodată, ca procedând astfel să ajutăm la familiarizarea cu una dintre cele mai interesante teorii gramaticale, teorie care, în momentul de față, este cunoscută în România aproape exclusiv de cei ce studiază limba din perspectiva informatică.

3. REPRESENTĂRI: TRĂSĂTURI, VALORI ȘI MATRICE

Orice teorie gramaticală studiază un anumit gen de entități gramaticale, sau, mai general spus, lingvistice, precum propoziții, fraze, funcții sintactice, părți de vorbire. Această preocupare caracterizează și HPSG. În HPSG, obiectele lingvistice sunt tratate într-un mod unitar, datorită unui sistem de notație suficient de general. Sistemul se bazează pe principiul funcțiilor matematice.

O funcție creează perechi de argumente și valori. Acest fapt este fructificat de către HPSG, în direcția codificării informației gramaticale. Fie, de pildă, faptul gramatical că un constituent oarecare are cazul acuzativ. Acest fapt va putea fi reprezentat în HPSG, cu ajutorul unui simbol pentru argument și cu ajutorul unui alt simbol pentru valoarea funcției. Argumentul va fi cazul, iar valoarea, acuzativul. Ceea ce în matematică se numește argumentul funcției, în unele teorii lingvistice poartă numele de *trăsătură* (sau *atribut*) și reprezintă analogul trăsăturilor fonetice distinctive.

În HPSG, trăsătura e reprezentată prin abrevieri cu majuscule: SUB(IECT), GEN, N(UMĂ)R etc. Așadar, CAZ va nota trăsătura exemplificată mai sus. Valoarea trăsăturii CAZ va fi o entitate reprezentată prin minuscule subliniate: ac(uzativ). Notația pentru perechea trăsătură-valoare va fi, așadar, CAZ: ac.

Orice obiect lingvistic este reprezentat printr-o structură de astfel de perechi trăsătură-valoare, care poate fi scrisă sub forma unei *matrice*. Matricele sunt reprezentările fundamentale (și unice!) în HPSG, tot așa cum arborii sunt reprezentările fundamentale în gramaticile transformaționale. O matrice arată după cum urmează:

$$M1 \begin{bmatrix} \text{GEN} & : \textit{masc} \\ \text{NR} & : \textit{sg} \\ \text{CAZ} & : \textit{ac} \end{bmatrix}$$

Această matrice reprezintă un obiect lingvistic care poate da seama, de pildă, de condițiile de acord.

Este, desigur, preferabil să cunoaștem și numele obiectului dotat cu proprietatea de a controla acordul. Denumirile cu minuscule subliniate, așezate înaintea matricei, servesc tocmai unor asemenea scopuri. Pentru situația de față, de exemplu, se impune o adăugire, ce semnalează că obiectul lingvistic descris mai sus intervine în controlul acordului și conține exact aceste trăsături, și numai pe acestea. Este așadar nevoie de un nume pentru matricea M1. O asemenea reprezentare (mai informativă decât M1) va fi M1':

$$M1' \quad \text{acord} \begin{bmatrix} \text{GEN} : \textit{masc} \\ \text{NR} : \textit{sg} \\ \text{CAZ} : \textit{ac} \end{bmatrix}$$

3.1. SPECII DE VALORI ALE TRĂSĂTURILOR DIN MATRICE

3.1.1. Valori atomice

Trăsături de tipul celor din M1 sunt foarte frecvente în HPSG. Caracteristica lor o constituie faptul că valoarea trăsăturii este o entitate ce nu se mai pretează unei descompunerii ulterioare, adică o entitate căreia nu i se mai poate asocia o matrice; masc, sg, ac sunt, prin urmare, socotite indecompozabile. Într-o asemenea situație, se spune că trăsăturile au valori atomice. Alte exemple de valori atomice sunt valorile trăsăturii PERS(OANĂ) (I, a II-a, a III-a). Tot atomice sunt valorile 'booleene' --simbolizate prin + și --.

3.1.2. Valori nonatomice

Este însă posibil ca valoarea unei trăsături să nu fie atomică. Fie, în acest sens, împrejurarea că dorim să precizăm că proprietatea de a participa la acord este a unui nume. În cazul acesta, trebuie adoptată o trăsătură corespunzătoare, care să indice categoria (partea de vorbire) nume. Vom introduce, în acest scop, trăsătura CAT(EGORIE). Pe de altă parte, e nevoie și de o trăsătură care va numi proprietatea propriu-zisă de a participa la acord; o vom simboliza cu ACORD. Valoarea acesteia va fi tocmai matricea M1' și va fi, prin urmare, o valoare nonatomică. Descrierea acestui obiect lingvistic va conduce la următoarea matrice:

$$M2 \left[\begin{array}{l} CAT : \textit{nume} \\ ACORD : \left[\begin{array}{l} GEN : \textit{masc} \\ NR : \textit{sg} \\ CAZ : \textit{ac} \end{array} \right] \end{array} \right]$$

M2 codifică exact ceea ce s-a dorit, dar mai rămâne să se precizeze ce fel de obiect este cel descris. Acest lucru se realizează prin adoptarea unui corespunzător, care, în situația de față, este *nume*.

3.1.3. Mulțimi și liste

O situație oarecum specială este reprezentată de împrejurarea în care valoarea unei trăsături este fie o listă, fie o mulțime. Diferența dintre liste și mulțimi este că în ultimele apar membri neordonați, în timp ce primele conțin membri ordonați. Listele se notează prin perechi de paranteze unghiulare (<...>), iar mulțimile sunt simbolizate prin perechi de acolade ({...}).

Este destul de evident de ce este nevoie să se apeleze la liste sau la mulțimi. Dacă, de pildă, se dorește să se arate care sunt complementele unui anumit verb (pentru conceptul de complement în HPSG, a se vedea mai jos, 4.1.1.), atunci valoarea trăsăturii corespunzătoare, COMP(LEMENT), va trebui să fie o l i s t ă , adică un ansamblu ordonat de asemenea entități. Ordonarea complementelor se realizează, în HPSG, pe criteriul oblicității lor crescânde față de centru (cf., de asemenea, 4.1.1.). În mod analog, dacă se dorește să se arate care este valoarea trăsăturii care specifică adjuncții unui verb (adică, în general, complementele circumstanțiale), atunci vom fi obligați să specificăm o m u l ț i m e de valori.

Situația listelor sau a mulțimilor este oarecum specială, în sensul că listele și mulțimile nu pot fi calificate apriori valori atomice sau nonatomice. Astfel, mulțimea adjuncțiilor, ca și lista complementelor unui verb sunt valori nonatomice, deoarece membrii acestora sunt obiecte lingvistice complexe. Dacă o listă sau o mulțime este vidă --fapt de asemenea posibil-- matricea în cauză va conține o valoare atomică. Mulțimile sau listele vide sunt reprezentate prin acolade/paranteze unghiulare între care nu se găsește nimic: { } , < > .O ilustrație de listă vidă este lista-valoare a trăsăturii SUB pentru verbul *a-i păsa(cuiva) de ceva/cineva*.

3.1.4. Valori structural identice

Există multe situații în care valorile a două submatrice dintr-o matrice oarecare sunt identice. Trebuie să facem însă distincția între situația în care submatricele sunt accidental identice și cea în care acestea sunt structural identice. Un exemplu de identitate accidentală apare în propoziția următoare:

(1) *El l-a văzut.*

Cele două pronume au aceeași persoană, gen și număr. Acest lucru nu este totuși decisiv pentru gramaticalitatea enunțului. Dimpotrivă, în exemplul:

(2) *Ion se gândește la vacanță.*

identitatea dintre persoana și numărul reflexivului și persoana și numărul subiectului este esențială. În acest din urmă caz, vorbim despre o identitate structurală.

Identitățile structurale se exprimă cu ajutorul unei tehnici specifice, numite coindexare. Coindexarea constă în atribuirea de indecși identici valorilor identice. Idecșii sunt numere naturale, care aici vor fi încadrate de bare verticale; de pildă, |1|.

O ilustrație: să presupunem că suntem în situația de a arăta că genul, numărul și cazul unui adjectiv sunt identice cu genul, numărul și cazul substantivului pe care adjectivul îl determină. Acest lucru va putea fi captat în sistemul de notație al HPSG mulțumită mecanismului coindexării (dar trebuie precizat că tratamentul acestui tip de acord, în HPSG, nu este, de fapt, cel indicat în matricea M3):

$$M3 \left[\begin{array}{l} ADJ : [ACORD : |1|] \\ NUME : ACORD : |1| \left[\begin{array}{l} GEN : masc \\ NR : sg \\ CAZ : ac \end{array} \right] \end{array} \right]$$

Mecanismul coindexării este util prin generalitatea sa. M3 codifică un fenomen pentru care gramaticile au un concept –acordul-, însă asemenea identități structurale sunt mult mai numeroase, iar nume pentru ele, în mod curent, nu există. Este, de pildă, la fel de important să se precizeze că în enunțuri precum:

(3) *Ion nu poate să primească pe nimeni azi.*

(4) *Ion nu poate primi pe nimeni azi.*

conjunctivul și infinitivul au același subiect ca și verbul regent. În mod similar, identitatea dintre genul, numărul, cazul și persoana unui complement direct și genul, numărul, cazul și persoana pronumelui personal neaccentuat care poate anticipa/relua complementul este esențială pentru gramaticalitatea enunțului. Toate aceste situații cad, în HPSG, sub incidența aceleiași proceduri -- coindexarea.

3.2. CLASIFICAREA REPREZENTĂRIILOR

În paragraful precedent s-au făcut referiri la numele matricelor. Aceste nume sunt denumite *tipuri*. Relația dintre obiectele lingvistice și tipuri este deja evidentă: orice obiect lingvistic este de un anumit tip.

Tipurile sunt supuse unei operații de clasificare, care, în HPSG, are o importanță specială. De felul cum se face clasificarea depinde plauzibilitatea, eleganța sau chiar reușita unei explicații gramaticale.

Correspondentul aproximativ, în descrierile gramaticale tradiționale, al tipurilor din HPSG sunt părțile de vorbire, părțile de propoziție și echivalentele propoziționale ale ultimelor. În termeni mai preciși, intersecția dintre mulțimea entităților identificate în gramaticile tradiționale și mulțimea tipurilor din HPSG este o mulțime nevidă.

Asemănările se opresc însă aici. Terminologia HPSG este relativ distinctă de cea tradițională. De asemenea, este diferită dispunerea entităților gramaticale într-o ierarhie. În gramaticile tradiționale, ierarhiile există, dar sunt parțiale. Pronumele demonstrativ de apropiere, spre exemplu, este o subspecie a pronumelui demonstrativ, care, la rândul său, se subsumează pronumelui. Dar,

în mod curent, ierarhia nu e completă, căci nu se știe, mai departe, care este raportul dintre pronume și substantiv. Este adevărat, această imprecizie e de multe ori corectată, prin recursul la categoria numelui, o categorie supraordonatoare atât în raport cu substantivul, cât și cu pronumele. Problema este totuși că descrierile tradiționale nu sunt interesate de o ierarhizare exhaustivă a "tipurilor" lor lingvistice. Pentru HPSG însă, exhaustivitatea ierarhizării este un obiectiv urmărit sistematic și declarat ca atare. Ierarhizarea este de tip piramidal.

Vom prezenta în continuare câteva elemente ale acestei clasificări. În vârful ierarhiei se găsește tipul semn. Acest tip poate fi considerat cel mai general posibil. Tipul semn se subdivide în alte două: tipul-s(emn)-sintagm(atic) și tipul s(emn)-nonsintagm(atic). Ultimul poate fi, la rândul său, lex(ical) sau nonlex(ical). Tipul lex constă în următoarele subtipuri: nume, verb, adjectiv, prep(oziție), adverb. Tipul nonlex se subdivide și el în tipurile afix(al) și nonafix(al). Tipul afix ar putea fi conceput ca fiind realizat de subtipurile afix-pronom(inal), afix-refl(exiv), afix-verb(al) și afix-adv(erbial). Această ultimă clasificare este, desigur, doar o propunere. Ea s-ar putea dovedi avantajoasă, ținându-se seamă de faptul că formele neaccentuate ale pronumelui personal (afix-pron), ale celui reflexiv (afix-refl), formele de auxiliar ale perfectului compus, ale condiționalului și ale viitorului (afix-verb), precum și acel mic grup de adverbe care sparg unitatea formelor verbale compuse --și anume adverbele *și*, *mai*, *cam*, *tot*-- (afix-adv) au un comportament mai degrabă "morfematic".

Tipul nonafix, la rândul său, s-ar putea reprezenta prin subtipurile marc(ator) și det(erminator); marc are ca subtipuri conj(unctie) și c(om)pl(emen)t(izor).

Clasificarea obiectelor lingvistice înaintază, în modul schițat mai sus, până la baza piramidei, adică până la punctul în care nici o clasificare nu mai este relevantă pentru problema corectitudinii gramaticale.

Rămâne de subliniat o proprietate a acestei ierarhizări: orice obiect care este de un anumit tip este, în același timp, de oricare alt tip dintre cele care îl domină. Fie, de pildă, obiectul de tipul "verb cu afix pronominal în acuzativ", unul dintre posibilele tipuri necesare unor explicații gramaticale (exemplificat prin *a-i vedea*); acest obiect va fi, în același timp, și de următoarele tipuri: "verb cu afix pronominal" (care, în mod plauzibil, este tipul imediat superior), "verb", "lexical" și, în sfârșit, "semn". Evident, el nu va fi de *tipul* "semn sintagmatic", deoarece ierarhia schițată anterior nu permite să se ajungă la tipul "verb cu afix pronominal în acuzativ", trecându-se prin ramura sintagmelor. Așadar, posibilitatea unui obiect de un anumit tip de a fi, în același timp, și de alte tipuri, decurge din faptul că ierarhia este organizată pe principiul implicației, a cărei proprietate logică este tranzitivitatea.

Prezentarea ierarhiei tipurilor poate fi aici întreruptă, urmând a fi detaliată atunci când componentele HPSG vor fi trecute în revistă. Ceea ce este important de subliniat este faptul următor: acest fel de reprezentări și de clasificări se întâlnesc pretutindeni în teritoriul HPSG. Ele sunt, pentru a spune așa, tehnologia de construcție a teoriei în ansamblul ei. Rămâne acum de studiat ansamblul arhitectonic al teoriei din punctul de vedere al blocurilor de construcție. Acestea sunt:

- (i) lexiconul;
- (ii) a) regulile sintagmatice,
b) constrângerile universale asupra acestor reguli;
- (iii) regulile de precedență liniară.

4. LEXICONUL

4.1. INTRĂRI LEXICALE

În această componentă se găsesc reprezentări ale semnelor nonsintagmatice. Clasificarea acestor semne a fost deja făcută în paragraful precedent. Specific HPSG este însă faptul că se atribuie intrărilor lexicale o structură internă, reprezentată, desigur, matricial.

4.1.1. Trăsăturile FON și SINSEM

Cele mai generale trăsături care intervin în descrierea structurii intrărilor lexicale sunt trăsăturile FON și SINSEM. Aceste trăsături nu caracterizează doar semnele nonsintagmatice, dar în acest paragraf numai ele vor fi luate în considerație.

În HPSG, se consideră că orice semn este deținător de informație fonologică (FON) și sintactico-semantică (SINSEM). HPSG are disponibilitatea de a codifica și o parte a informației pragmatice deținute de unele dintre semne, însă acest lucru nu va putea fi discutat aici; pentru toate acestea, a se vedea Pollard și Sag 1987:81-112, Pollard și Sag 1994:27.

Nu vom insista asupra valorii trăsăturii FON. Trăsătura SINSEM este concepută ca relevând două categorii de informație: locală (loc) și nonlocală (nonloc). Informația locală indică variate proprietăți gramaticale, semantice și pragmatice, cum ar fi: cu ce fel de semn avem de-a face, ce conținut are, care îi sunt disponibilitățile de combinare. Ea este, în mod mai specific, codificată prin trăsăturile CAT(EGORIE), VAL(ENȚĂ), CONT(CONȚINUT) și C(ON)T(E)X(T).

CAT este o trăsătură prin intermediul căreia este precizată "apartenența

categorială". Fie un nume oarecare. O parte din matricea corespunzătoare va conține informația că semnul în cauză are *această* apartenență categorială, și nu alta:

M4 *cat* [CAT: *centru* [CENTRU: *nume*]]

Trăsătura VAL servește la indicarea particularităților de combinare. Este vorba despre acele combinații în care obiectul în cauză –pentru cazul de mai sus, de tipul *nume*-- joacă rolul de centru.

Valoarea corespunzătoare trăsăturii VAL este obiectul din următoarea structură:

$$M5 \text{ loc } \left[\begin{array}{l} \text{CENTRU : } \textit{nume} \\ \text{CAT : VAL : } \left[\begin{array}{l} \text{COMP : ...} \\ \text{SUB : ...} \\ \text{SPR : ...} \\ \text{MARC : ...} \\ \text{AF : ...} \end{array} \right] \\ \text{OBLIC - ORD - VAL : ...} \end{array} \right]$$

M5 arată, așadar, că valențele unui centru se prezintă sub forma unui spectru de posibilități de combinare. Spectrul este indicat prin trăsăturile COMP, SUB, SPR, MARC (cel puțin pentru unele elemente din limba română) și AF.

Prin COMP, este exprimată disponibilitatea de combinare a unui centru cu complementele sale. Conceptul de complement, în HPSG, este înțeles într-un mod parțial diferit de felul în care este definit în gramatica tradițională. Complement poate fi, de exemplu, un constituent care este în dependență rețională de un alt constituent. Complementele circumstanțiale din descrierile

tradiționale nu ar putea fi, așadar, considerate complemente în accepția HPSG, deoarece ele nu întrețin o asemenea relație de dependență cu centrul.

În al doilea rând, complementul, în HPSG, nu este dependent doar de verb, ci de orice obiect față de care manifestă dependență recțională. Aceasta înseamnă că, în HPSG, se vor găsi complemente ale numelui, verbului, ale prepoziției sau ale adjectivului. În legătură cu numele, se cuvine însă a fi precizat faptul că, date fiind slabele capacități de recțiune ale acestuia, drept complemente trec acei constituenți care sunt doar *semantic* complemente. Aceasta înseamnă că în HPSG (dar și în alte teorii gramaticale de dată recentă) valorile semantice ale genitivului nominal joacă un rol mai mare decât în analizele tradiționale.

Trăsătura SUB codifică posibilitatea unui centru de a avea subiect. Și conceptul de subiect este gândit într-un mod mai larg, deoarece această trăsătură de valență poate caracteriza, în HPSG, nu doar verbul, ci și numele (sau adjectivul). Subiectul unui nume, de pildă, corespunde valorii subiective a genitivului, în grupuri nominale precum *venirea lui Ion sau lupta papei împotriva comunismului*.

Disponibilitatea unui constituent-centru de a avea afixe este codificată prin trăsătura AF. Nu toate afixele se bucură însă de aceeași atenție în HPSG. De pildă, morfemele persoanei verbului nu vor fi luate în considerație (chiar dacă ele sunt afixe), spre deosebire de formele neaccentuate ale reflexivului. Explicația se găsește în faptul că prezența ultimelor produce efecte considerabile asupra structurilor mai largi (aici, structurile sintagmatice). Aceste efecte sunt studiate de o altă componentă a HPSG: teoria dependențelor anaforice. Dimpotrivă, nimic asemănător nu urmează din folosirea afixului de persoană. Ca regulă informală, s-ar putea așadar spune că

afixe demne de atenție în HPSG sunt cele a căror întrebuințare antrenează efecte dincolo de limitele unităților lexicale în care apar.

SP(ECIFICATO)R este o trăsătură fără corespondent univoc în gramatica tradițională. Cu ajutorul ei se intenționează să se capteze, în interiorul unui grup, relații mai complexe decât simpla dependență. În mod intuitiv, specificatorii sunt articole, pronume și adjective nehotărâte, numerale cardinale și ordinale, articole, pronume și adjective demonstrative, genitive posesive pronominale și nominale sau, în sfârșit, pronume și adjective relative (despre care trebuie însă precizat că Pollard și Sag le interpretează drept nume).

Specificatorii sunt deținătorii trăsăturii SPEC(IFICAT). Această trăsătură caracterizează în primul rând un anumit tip, tipul det(erminator). După cum se va vedea însă imediat, tipul det nu este singurul posesor al trăsăturii SPEC.

Valoarea acestei trăsături este centrul construcției. Se înțelege de aici că raportul dintre specificator și centrul său este un raport mai complex. Nu numai centrul își selectează specificatorul, ci și invers. Aceasta pare să demonstreze că relația dintre centru și specificator este mai slabă decât cea dintre centru și complemente sau subiect.

Este de notat că în exemplele date de Pollard și Sag nu apar specificatori care "să selecteze" un centru verbal.

Trăsătura MARC(AJ) codifică o proprietate specifică unui anumit tip lingvistic. Este vorba de tipul marc(ator). Prin urmare, MARC nu poate apărea ca trăsătură a unui obiect de alt tip. Despre rolul acestei trăsături se va putea vorbi mai clar când va fi luat în discuție principiul care reglementează comportamentul acesteia (cf. *infra*, 5.3.4.).

Orice constituent care deține trăsătura MARC deține, de asemenea, și trăsătura SPEC(IFICAT), dar reciproca nu e adevărată. Acesta este un alt mod de a spune că entitățile de tipul marc s-ar putea caracteriza prin următoarea reprezentare matricială:

$$M6 \text{ marc} \begin{bmatrix} \text{MARC : ...} \\ \text{SPEC : ...} \end{bmatrix}$$

Ca și în cazul trăsăturii SPEC, valoarea trăsăturii MARC este centrul construcției.

O precizare importantă în legătură cu valorile trăsăturilor de valență discutate mai sus se găsește în Pollard și Sag 1994:23. Se afirmă acolo că aceste valori nu sunt obiecte de tipul semn, ci de tipul sinsem. Obiectele de acest din urmă tip se caracterizează prin aceea că nu încorporează trăsătura FON. Ele codifică informația sintactică și semantică deținută de un semn și sunt, așadar, numai ingrediente ale semnelor.

Acest mod de a pune problema este important deoarece permite păstrarea unei anumite generalități, absolut necesare atunci când sunt luate în discuție valențele unui centru. Prezența trăsăturii FON ar fi obligat la referiri concrete.

Rămâne acum să se explice rostul trăsăturii [OBLIC-ORD-VAL:...] din M5. Ea codifică informația referitoare la ordinea de *oblicitate* a valențelor centrului. Prin intermediul acestei trăsături este, așadar, specificată o listă a valențelor, ordonate în raport cu gradul crescător de oblicitate pe care ele îl manifestă față de centru.

Primul de pe listă va fi constituentul cel mai puțin oblic, adică subiectul; al doilea va fi specificatorul, urmat de complemente. Grupa complementelor presupune ea însăși o (sub)ierarhie a oblicității, obiectul direct fiind cel mai

puțin oblic, cel indirect fiind mai oblic decât cel direct ș.a.m.d. Afixele reflexive și de pronume personal ar trebui și ele să fie incluse în această ierarhie.

Trebuie totuși precizat că HPSG nu propune o teorie propriu-zisă a oblicității. Pollard și Sag invocă doar anumite fenomene (de pildă, ordinea constituenților, dependențele anaforice, ordinea de saturare a centrilor de către valențele lor). Ei abstrag de aici ipoteza --de mare importanță pentru întreaga construcție a teoriei-- conform cu care relațiile gramaticale ar fi guvernate de o anumită ierarhie. Aceasta este ierarhia de oblicitate.

Dar întrebarea firească este la ce folosește codificarea acestui gen de informație. Răspunsul e că, în HPSG, ea reprezintă baza fenomenelor mai înainte amintite. Voind să evite explicațiile gramaticale în termeni configuraționali --așa cum se procedează, de pildă, în Teoria G&B (= Government and Binding) -, Pollard și Sag recurg la conceptul de oblicitate, care vizează relații de natură lexicală.

4.1.2. Trăsătura NONLOC

Fie următoarele exemple:

(5) *Tablouri ca acestea, Muzeul Național nu va achiziționa niciodată.*

(6) *Muzeul Național nu va achiziționa niciodată tablouri ca acestea.*

Diferența dintre (5) și (6) constă în topicalizare. Obiectul direct din (6:) are locul său normal, în timp ce în (5) același obiect direct se găsește la început de propoziție. Dar faptul că această sintagmă este recunoscută în continuare drept obiect direct al verbului dovedește că ea nu a încetat să-și păstreze relația recțională cu centrul. Afectată este doar relația de vecinătate cu acesta. În limbi cu topică mai rigidă decât româna astfel de fenomene sunt mai acut resimțite.

Pentru explicarea structurilor de tipul (5), HPSG angajează trăsătura NONLOC(AL); această trăsătură este destinată să dea seama de dependențe la distanță. În acest sens, se propune să se pornească de la situații precum (6). Verbul *a achiziționa* din (6) are următoarea reprezentare în matrice (relativ la ceea ce interesează în discuția de față):

$$M7 \text{ sinsem} \left[\begin{array}{l} \text{LOC} : \text{loc} \left[\text{CAT} : \text{cat} \left[\begin{array}{l} \text{CENTRU} : \text{verb} \\ \text{COMP} : |1| < \text{obdir} > \end{array} \right] \right] \\ \text{NONLOC} : \text{abs}[\text{ABS} : \{\}] \end{array} \right]$$

M7 arată că *a achiziționa* este un verb a cărui listă de complemente e constituită dintr-un complement direct. Submatricea [*abs...*] arată că verbul nu este posesorul nici unei trăsături nonlocale; nu ar trebui deci să ne așteptăm ca el să facă parte dintr-o structură 'dislocată' precum (5). Printr-o regulă lexicală, care va fi prezentată alături de altele de același fel mai jos (vezi 4.2.), se obține acum o variantă a verbului parțial descris în M7. Această variantă este următoarea:

$$M8 \text{ sinsem} \left[\begin{array}{l} \text{LOC} : \text{loc} \left[\text{CAT} : \text{cat} \left[\begin{array}{l} \text{CENTRU} : \text{verb} \\ \text{COMP} : < > \end{array} \right] \right] \\ \text{NONLOC} : \text{abs}[\text{ABS} : \{|1|\}] \end{array} \right]$$

M8 indică faptul că, spre deosebire de descrierea din M7, varianta lui *a achiziționa* se caracterizează printr-o valoare nonvidă pentru trăsătura NONLOC. Matricea care este valoarea acestei trăsături se numește abs(ent); ea e constituită, la rândul ei, din trăsătura '(constituent) ABS(ENT)' cu valoarea obiect direct; acesta este obiectul direct care figurează în lista de complemente

din matricea M7. Identitatea este semnalată prin coindexare și e de observat că lista de complemente din M8 nu mai conține nimic. Prin urmare, informația codificată în M8 este că verbul parțial descris acolo se caracterizează printr-un obiect direct absent. Este astfel evident că trăsătura NONLOC reprezintă premisa de existență a sintagmelor cu centru și constituent antepus, deoarece constituentul-lipsă urmează să se regăsească altundeva în enunț. De modul în care el ajunge să apară în alt punct al enunțului nu mai este însă răspunzătoare trăsătura NONLOC.

Ceea ce mai trebuie să se adauge este că trăsătura ABS nu e, în HPSG, singura care precizează informația de ordin nonlocal. Tot purtătoare de informație nonlocală mai sunt considerate trăsăturile INT(EROGATIV) și REL(ATIV), adică tocmai acele trăsături care caracterizează pronumele corespunzătoare.

4.1.3. Adjuncți

Adjuncții nu fac parte din domeniul de valențe ale centrului. Cu toate acestea, relația unui adjunct cu centrul nu este o relație de simplă compatibilitate. Se consideră astfel că un adjunct își selectează centrul. Faptul acesta este codificat prin trăsătura MODIF(ICAT), a cărei valoare este, desigur, centrul.

Se observă că MODIF joacă același rol ca SPEC. În aceste condiții, este necesară sublinierea diferenței dintre ele.

Această diferență nu e însă atât de ușor de pus în evidență și acesta pare a fi unul dintre punctele slabe în HPSG. Diferența ar putea fi de natură semantică. Adjuncții sunt centrii semantici ai unei sintagme. Specificatorii, nu. Diferența se poate sesiza prin următoarele exemple:

(7) *Ion se plimbă agale.*

(8) *Câțiva soldați.*

În (7), *agale* este adjunct față de sintagma *Ion se plimbă*, care este centru. Dar, din punct de vedere semantic, cel care este centrul este *adjunctul*, deoarece, în perspectiva distincției argument/funcție, nu centrul sintactic este funcție, ci adjunctul. Într-adevăr, *agale* are ca argument semantic sintagma - centru *Ion se plimbă*, fiind de fapt o predicatie despre starea de lucruri descrisă de propoziție.

Tocmai acest lucru nu se întâmplă în relația dintre *câțiva* și *soldați*, unde primul termen este specificatorul numelui-centru. În acest caz, determinantul rămâne secundar, inclusiv din punct de vedere semantic, în raport cu centrul. E adevărat, dependența semantică a specificatorului de centru nu se mai poate susține în termenii distincției funcție/argument, pentru că determinantul este un cuantificator, iar cuantificatorii nu sunt nici funcții, nici argumente. Cu toate acestea, ideea că numele este centrul semantic al construcției poate fi susținută invocând argumentul că specificatorul introduce o aproximare a unui conținut ce aparține numelui.

Ca exemple de adjuncti, se pot cita complementele circumstanțiale și echivalentele lor propoziționale din gramatica tradițională, gerunziile circumstanțiale, unele din tradiționalele elemente predicative suplimentare (ca, de pildă, *obosit*, în *Ion s-a întors obosit*), dar și nume genitivale (sau în acuzativ prepozițional) cu semnificație circumstanțială: *casele de pe deal*, *sportivul anului*, *oamenii nordului*. Pollard și Sag consideră, de asemenea, că determinările adjectivale ale numelui (de exemplu, *o carte interesantă*) sunt tot adjuncti.

4.2. REGULI LEXICALE

Un prim exemplu de regulă lexicală a putut fi întâlnit în discuția din jurul trăsăturii NONLOC. Regula constă în punerea în relație a două tipuri lexicale diferite: unul este tipul tranz(itiv)-verb (ilustrat de verbul *a achiziționa* din propoziția (6)); celălalt este un tip a cărui particularitate principală este faptul că valența de obiect direct este satisfăcută, în ciuda faptului că obiectul direct este absent. Să numim noul tip tranz-el(iptic)-verb. El deține o trăsătură nouă, pe care tipul tranz-verb nu o conținea: NONLOC.

Se poate pune acum întrebarea la ce folosesc aceste reguli. Răspunsul e că ele elimină repetițiile. Mulțumită regulii lexicale de mai sus, de exemplu, tipul tranz-el-verb nu are nevoie să mai fie detaliat în continuare în subtipurile care ar trebui să fie invocate dacă regula lexicală nu ar fi adoptată. Formele sale flexionare sunt deja formele flexionare ale tipului tranz-verb.

Forma unei asemenea reguli este, în termeni informali, următoarea:

Dacă există tipul *x*, atunci există de asemenea tipul *y*.

Regulile lexicale sunt folosite pe o scară largă în HPSG. Tratatamentul construcțiilor pasive, de exemplu, își are originea într-o astfel de regulă care acționează tot asupra tipului tranz-verb. Rezultatul ei este punerea în corespondență a acestui tip cu tipul part(icipiu)-pas(iv). Noul tip devine ulterior complementul verbului *a fi*.

5. COMPONENTA SINTAGMATICA

5.1. TIPURI (REGULI) SINTAGMATICE

În Pollard și Sag 1994:396, sunt menționate două mari subtipuri sintagmatice: tipul s-sintagm-c și tipul s-sintagm-non-c. Primul acoperă

totalitatea sintagmelor cu centru, în timp ce al doilea se referă la sintagmele lipsite de centru.

Ultimul tip are ca subtip s-sintagm-coord. El definește sintagmele cu constituenți coordonați. În mod doar aparent surprinzător, acestea nu sunt socotite sintagme cu centru; pentru argumente, a se consulta Sag, Gazdar, Wasow și Weisler 1988.

Celălalt tip constă în următoarele subtipuri: c-compl (sintagmă cu centru și complement(e)), c-sub (sintagmă cu centru și subiect), c-marc (sintagmă cu centru și marcator), c-spr (sintagmă cu centru și specificator), c-antep (sintagmă cu centru și constituent antepus), c-adjunct (sintagmă cu centru și adjunct).

Aceste subtipuri ar putea fi socotite analogul structurilor sintactice în sens larg. Analogia are însă doar rol euristic. În HPSG nu este în nici un fel admisă existența structurilor sintactice "în sine". Aceasta este o importantă deosebire între adepții HPSG și cercetătorii de orientarea lui Noam Chomsky. Concepția care colorează inventarul de tipuri sintagmatice de mai sus este că unele dintre acestea (și anume: c-compl, c-sub, c-spr și c-marc) sunt consecințele trăsăturilor de valență manifestate de către centrul sintagmei. Iată câteva ilustrații, cu marcarea constituentului noncentral:

- (i) c-compl: ...*îl vede pe Ion*
- (ii) c-sub: ***Ion** doarme*
- (iii) c-marc: ...*că Ion nu mai vine*
- (iv) c-spr: *casa **vecinului***
- (v) c-antep: ...***pe care** Ion l-a văzut*
- (vi) c-adjunct: ...*se plimbă **în parc***

În opinia lui Pollard și Sag, cele șase tipuri epuizează structurile sintagmatice cu centru din limbile care, precum româna, nu ilustrează

fenomenul de inversiune dintre auxiliar și subiect.

5.2. STRUCTURA SEMNELOR SINTAGMATICE: TRĂSĂTURA RAM

Toate trăsăturile întrebuițate în descrierea structurii interne a semnelor nonsintagmatice se regăsesc și în reprezentarea structurii celor sintagmatice. Ceea ce apare în plus este trăsătura RAM(URĂ), care se alătură celei de LOC și (eventual) NONLOC.

Pentru semnele sintagmatice cu centru, valorile lui RAM sunt următoarele: ram-c(ramura-centru), ram-sub(iect), ram-compl(ement), ram-antep(us), ram-marc(ator), ram-sp(ecificato)r, ram-adjunct.

Sub o altă formă, simbolurile pentru subiect, complement, specificator și marcator au apărut și în enumerarea trăsăturilor de valență, în matricea MS. Repetarea lor aici ca valori ale trăsăturii RAM nu este totuși redundantă. Diferența este următoarea: trăsăturile de valență enumerate în matricea MS exprimă, pentru a spune așa, în *principiu* valențele unui centru. Existența de principiu a valențelor se precizează, în HPSG, într-un mod specific. Valorile trăsăturilor sunt cuprinse în liste (a se vedea și exemplele din M7 și M8). De pildă, disponibilitatea unui verb de a avea un obiect direct este indicată astfel:

$$M9 \quad \text{sinsem} \left[\text{CAT} : \text{cat} \left[\begin{array}{l} \text{CENTRU} : \text{verb} \\ \text{VAL} : \text{comp}[\text{COMP} : < \text{obdir} >] \end{array} \right] \right]$$

Când însă verbul și-a realizat această disponibilitate, el încetează a mai fi un semn lexical, deoarece, împreună cu complementul corespunzător, formează un semn sintagmatic (în cazul de față, de tipul c-comp) " Conform convenției că acest semn deține o trăsătură suplimentară (RAM), el va avea (pentru situația

discutată aici) următoarea structură (pentru valorile lui RAM dăm o descriere simplificată):

$$M10 \quad c-comp \left[\begin{array}{l} SINSEM | LOC : loc \left[CAT : cat \left[\begin{array}{l} CENTRU : |1| \\ VAL : comp[COMP : \langle \rangle] \end{array} \right] \right] \\ RAM : ram \left[\begin{array}{l} R-C : ram-c \left[SINSEM | LOC : loc \left[CAT : cat \left[\begin{array}{l} CENTRU : |1| verb \\ VAL : comp[COMP : \langle obdir \rangle] \end{array} \right] \right] \right] \\ R-COMP : ram-comp[SINSEM | LOC | CENTRU : nume] \end{array} \right] \end{array} \right]$$

De observat că M9 se regăsește în M10 ca submatricea care descrie obiectul lingvistic de tipul ram-c și că, spre deosebire de M9, în M10 valoarea trăsăturii COMP a sintagmei este o listă vidă ($\langle \rangle$); în HPSG, aceasta înseamnă că centrul este *saturat* (desigur, în raport cu trăsătura de valență care are drept valoare lista vidă). Dacă în M10 valoarea lui COMP ar fi fost identică cu valoarea lui COMP din M9, prin aceasta s-ar fi arătat că semnul sintagmatic nu și-a satisfăcut disponibilitatea pentru *obdir*. Într-o asemenea împrejurare, în HPSG se vorbește despre semne *nesaturate*.

O precizare crucială se impune în legătură cu aceste semne sintagmatice: ele descriu, de fapt, reguli sintagmatice. Afirmția este paradoxală doar la prima vedere. În acest sens, e de reamintit forma generală a regulilor de rescriere din versiunile recente ale gramaticii generative:

RS (1) $X'' \rightarrow Spec, X'$

(2) $X' \rightarrow X_0, Comp.$

RS(1) afirmă că proiecția maximală a constituentului X se rescrie ca proiecția lui minimală plus specificatorul) iar RS(2) spune că proiecția minimală a lui X este constituentul lexical X^0 plus complementele sale. Ceea ce RS(1)-(2) afirmă poate fi exprimat și cu ajutorul tipurilor sintagmatice,

datorită ierarhizării lor. Diferența constă în notație; regulile sintagmatice în HPSG se suplimentează reprezentărilor matriciale.

5.3. PRINCIPII

Deși tipurile sintagmatice sunt reguli, ele nu dau seamă în mod complet de fenomenul gramaticalității. Câteva alte aspecte ale acestora rămân nereglementate. Aceasta explică în HPSG recursul la *principii*.

Principiile sunt un complement al reglementărilor deja amintite. Ele exprimă o altă latură a efortului de a arăta că gramaticalitatea este dependentă de un număr limitat de factori --uneori acționând izolat, dar cel mai adesea acționând prin cooperare.

Principiile reglementează alte aspecte ale distribuției obiectelor lingvistice. Vom prezenta, în cele ce urmează, doar o parte dintre ele, și anume: Principiul Trăsăturilor Centrale, Principiul Dominanței Imediate, Principiul Valenței, Principiul Trăsăturilor Nonlocale, Principiul Marcajului, Principiul Specificatorului, Principiile Dependenței Anaforice (pentru prezentarea lor exhaustivă, cf. Pollard și Sag 1994:399-401). Relevanța fiecăruia va fi testată în același fel: se vor urmări consecințele faptului că principiul în cauză este ignorat.

5.3.1. Principiul Dominanței Imediate (PDI)

PDI face necesară referirea prealabilă la conceptul de *schemă de dominanță imediată*. Acest concept a fost folosit, fără a fi însă numit, când au fost inventariate genurile principale de sintagme. De fapt, aceste genuri sunt scheme de dominanță imediată. În aceste condiții, PDI se formulează după cum urmează:

PDI Orice obiect lingvistic de tipul "sintagmă cu centru" satisface una și numai una dintre schemele de dominanță imediată.

Pentru a-i identifica relevanța, principiul poate fi ignorat. Consecința va fi că un semn sintagmatic cu centru va putea fi considerat ca satisfăcând simultan cel puțin una din schemele de dominanță imediată sau nici una. În felul acesta, nu va mai exista nici un temei pentru a spune că o anumită sintagmă este de tipul "centru-adjunct", "centru-complement" etc. Prin dispariția PDI, este amenințată, așadar, însăși clasificarea semnelor sintagmatice.

Renunțarea la PDI atrage după sine și imposibilitatea de a explica incorectitudinea unor enunțuri precum:

(9) * *Problema constă.*

(10) * *Ion se comportă.*

În ambele cazuri, constituenți verbali conțin informații precise referitoare la complementele lor. Aceste informații obligă verbele în cauză să se insereze în anumite structuri sintagmatice a căror natură este semnalată de însuși inventarul de valențe ale verbelor; este vorba despre sintagme de tipul c-compl. Numai că nici (9), nici (10) nu sunt structuri sintagm(atice) în care să poată fi identificat tipul c-compl. Într-adevăr, acest tip reclamă existența unei valori ram-compl *nonvide*, ceea ce nu se întâmplă cu exemplele (9)-(10).

5.3.2. Principiul trăsăturilor centrale (PTC)

Acest principiu se exercită asupra obiectelor de tipul centru, menționat de mai multe ori în exemplificările anterioare. De aceea, domeniul său de acțiune este clasa semnelor sintagmatice cu centru, privite într-o dublă perspectivă: de sus în jos --adică de la semnul sintagmatic către subconstituenții săi-- și de jos în

sus, adică de la sintagmă spre o sintagmă mai complexă, în care cea dintâi se inserează.

PTC se exprimă astfel:

PTC. În orice semn sintagmatic cu centru, valoarea trăsăturii CENTRU este structural identică cu valoarea trăsăturii CENTRU a subconstituentului de tipul ram-c al sintagmei respective.

Ca o ilustrare a funcționării PTC, fie sintagma *soare strălucitor*. Reprezentarea ei matricială (simplificată la ceea ce interesează aici și în acord cu PTC) va fi următoarea:

$$M11$$

$$c - adjunct \left[\begin{array}{l} \text{SINSEM} \mid \text{LOC} \mid \text{CAT} : [\text{CENTRU} : |1|] \\ \text{RAM} : \left[\begin{array}{l} \text{R} - \text{C} : \text{ram} - c[\text{CENTRU} : |1| \mid \text{nume}[\text{N} : \text{caz}[\text{CAZ} : \text{nomin} \cup \text{ac}]]] \\ \text{R} - \text{ADJCT} : \text{ram} - adjunct[\text{CENTRU} : \text{adjectiv}[\text{MODIF} : |1|]] \end{array} \right] \end{array} \right]$$

M11 ilustrează PTC prin aceea că, pentru valoarea trăsăturii CENTRU, sintagma *soare strălucitor* moștenește valoarea aceleiași trăsături deținute de subconstituentul *soare*, descris de obiectul de tipul ram-c.

PTC este acel principiu în virtutea căruia se poate susține că, dacă, de pildă, o sintagmă este declarată nominală, atunci centrul ei nu poate fi decât un nume. Deși o asemenea afirmație apare mai degrabă ca fiind de la sine înțeleasă, ea nu este deloc astfel. Sau --ceea ce este același lucru-- măsura în care ea apare ca evidentă demonstrează cât de importantă este. Să presupunem, în acest sens, că PTC nu este asumat ca principiu. Respectarea lui nu va fi, în consecință, obligatorie. În această alternativă, sintagma *soare strălucitor* va fi în continuare caracterizată ca sintagmă de tipul c-adjunct --căci, în mod evident, neglijarea PTC nu implică o identificare a sintagmei cu un alt tip.

Numai că, în absența constrângerii exprimate prin PTC, sintagma va putea primi trei reprezentări, în egală măsură legitime: una identică cu M 11, alta în care centrul este adjectivul și alta în care atât adjectivul, cât și numele vor fi declarați centri. Această situație este însă intolerabilă. Ea înseamnă, de fapt, că termeni precum "centru" și "adjunct" devin inutili, pentru că își pierd însăși rațiunea pentru care au fost introduși. Prin extensie, concepte precum "specificator", "marcator", "constituent antepus" etc. suferă aceeași gravă devalorizare.

PTC este un principiu al gramaticii universale, dar utilizarea sa în raport cu o limbă dată reclamă cercetări specifice. Se întâmplă așa, deoarece numai astfel de cercetări pot arăta ce trăsături pot conta într-o limbă dată, ca trăsături ale centrului.

5.3.3. Principiul Valenței (PV)

Acest principiu este enunțat în capitolul al nouălea din Pollard și Sag 1994 și vine să înlocuiască Principiul Subcategorizării, asumat de cei doi autori, atât în Pollard și Sag 1987, cât și în primul capitol al cărții lor din 1994.

Felul în care este formulat *PV* este mai complicat decât înțelesul său intuitiv. Este necesară, mai întâi, introducerea a două noțiuni preliminare; simbol T_{val} și noțiunea de ramuri-surori.

T_{val} este un simbol definit pe mulțimea trăsăturilor de valență COMP, SUB, SPR și, pentru limba română, MARC. În ce privește conceptul de ramuri-surori, el este prezent într-un mod implicit în enumerarea obiectelor de tipul ram făcută mai înainte: dacă, spre exemplu, avem o entitate de tipul c-compl, această entitate va avea, ca valori ale trăsăturii RAM, obiectele de tipul ram-c și ram-compl. Cele două sunt ramuri-surori.

PV Pentru orice obiect de tipul s-sintagm-c valoarea trăsăturii T_{val} este identică cu valoarea aceleiași trăsături aparținând obiectului de tipul ram-c, cu excepția acelor valori care saturează ram-c și care astfel devin constituenții-surori ai lui ram-c.

Să considerăm PV în relație cu verbul *a dăru*. Acest verb are două complemente și un subiect, ceea ce înseamnă că valorile trăsăturilor de valență COMP și SUB nu vor fi liste vide. Matricea corespunzătoare verbului *a dăru* aduce câteva informații importante:

(i) prin trăsăturile de valență, ea arată că acest verb deține capacitatea de a se insera într-o structură sintagmatică:

(ii) prin aceleași trăsături, ea arată că verbul poate deveni centrul construcției în care se inserează. În aceste condiții, verbul va constitui obiectul de tipul ram-c.

Să presupunem acum că inserarea verbului în structura sintagmatică se realizează, rezultatul fiind sintagma *dăruiește Ioanei flori*. Construcția cade sub incidența PTC, PDI și PV. În ce-l privește pe ultimul, acesta stipulează că sintagma va moșteni de la verbul-centru toate valorile trăsăturilor de valență, mai puțin cele care, fiind realizate ca ramuri, saturează, deja ramura-centru. Sintagma va poseda, așadar, valoarea vidă a trăsăturii COMP și valoarea *nonvidă* a trăsăturii SUB, căci valoarea acestei ultime trăsături nu s-a actualizat. Conceptele de saturare și nesaturare introduse în discuția din jurul matricei M10 își găsesc acum justificarea de utilizare.

Utilitatea PV este imediat observabilă. În absența sa, modul de transmitere a trăsăturilor de valență de la ram-c la sintagma căreia ram-c îi

aparține nu poate fi reglementat. În consecință, posibilitatea de apariție a unor sintagme precum **dăruiește Ioanei Ioanei flori flori*, **Ion Ion dăruiește Ioanei flori Ioanei flori* nu ar putea fi interzisă.

În ilustrația dată mai sus a rămas totuși nelămurit un lucru: nu a reieșit din ce motiv se consideră că verbul se inserează într-o sintagmă de tipul "centru--complement" și nu, de pildă, într-una de tipul "centru-subiect" (dat fiind faptul evident că verbul deține și trăsătura de valență SUB). Explicația se găsește în ierarhia centrilor pe care Pollard și Sag o propun (Pollard și Sag 1994:362). Această ierarhie provine din aceeași teorie a oblicității asumate în HPSG. Conform cu această ierarhie, ordinea de saturare a centrilor este ordinea de oblicitate descrescătoare a valențelor acestuia. Complementele sunt cele mai oblice și, prin urmare, ele vor fi primele care vor satura centrul. Specificatorii sunt mai puțin oblici, și vor satura centrul de care sunt dependenți numai după ce s-a efectuat saturarea cu complemente (în caz că acestea există). Subiectele saturează ultimele. Prin această ordine de saturare, Pollard și Sag afirmă că reconstruiesc prin principii lexicale Teoria Nivelelor de Proiecție Sintagmatică ("X-bar Theory").

5.3.4. Principiul Specificatorului și al Marcajului (PS și PM)

Aceste două principii reglementează folosirea trăsăturilor SPEC și MARC. SPEC este o trăsătură în virtutea căreia un determinant își selectează centrul. Principiul Specificatorului are drept consecință captarea într-un mod economic a unor fenomene de acord, atunci când specificatorul este posesor de suficientă informație gramaticală în acest sens (fiind, de pildă, un adjectiv pronominal). Datorită faptului că trăsătura SPEC are ca valoare centrul, determinantul va fi obligat să preia de la acesta toată informația pe care centrul o poate da --cu

condiția, desigur, ca determinantul însuși să fie capabil să preia această informație.

Absența acestui principiu ar obliga la introducerea unui simbol de acord care să semnaleze concordanța dintre determinant și centru. În principiu, acesta nu este un lucru dezavantajos. El e doar mai puțin economic. În general, e de dorit ca numărul simbolurilor întrebuințate să fie cât mai mic.

Un constituent posesor al trăsăturii MARC transmite această trăsătură sintagmei imediat superioare, dar nu mai sus de ea. Fie, de pildă, relația dintre conjuncția *că* și propoziția *Ion nu mai vine*, în cadrul propoziției subordonate *...că Ion nu mai vine*; *că* este interpretat ca marcator. În virtutea PM, el transmite prin urmare trăsătura MARC propoziției *că Ion nu mai vine*, care arată astfel că este o propoziție subordonată.

Principiul marcajului ar putea fi utilizat pentru limba română în legătură cu elemente precum *a* infinitival, conjuncțiile subordonate *că*, *să*, *dacă*, sau pentru *pe*, marcator al complementelor directe dublate.

5.3.5. Principiile Dependentei Anaforice (PDA)

Ca și alte teorii gramaticale de dată recentă, HPSG tratează în mod separat problemele anaforei. Le tratează separat, în măsura în care anumite cazuri de anaforă expun un comportament sistematic și sunt, în consecință, apte de a deveni obiectul unor principii din care se deduc explicații ale gramaticalității.

În HPSG, principiile care au ca obiect anaforicitatea au la bază patru categorii de elemente: existența anumitor tipuri semantice, dispunerea acestor tipuri în ordinea de oblicitate crescătoare (de la stânga la dreapta), definirea unei relații ce se stabilește între tipurile semantice ordonate după criteriul oblicității și mecanismul identității structurale (coindexarea).

(i) Tipurile semantice sunt obiectele lingvistice care pot apărea în câmpul

trăsăturii CONT (conținut). Vom studia aici doar câteva aspecte legate de conținutul semnelor de tipul nume deoarece acestea sunt cele care au legătură cu anaforicitatea gramaticală.

În Pollard și Sag 1994:249, se propune ca numelor să li se atașeze conținutul generic ob(iect)nom(inal). Acest tip are în continuare următoarele subtipuri: ob-nonpron, ob-pron(ominal).__Primul caracterizează conținutul substantivelor, în vreme ce celălalt e propriu conținutului pronomelor; ob-pron la rândul său, se divide în anaf(oric) și nonanaf(oric), iar primul are drept diviziuni refl(exiv) și rec(iproc).

Pentru limba română, este preferabil ca această clasificare semantică să fie folosită și în legătură cu afixele pronominale, dat fiind că și ele sunt purtătoare de asemenea conținuturi.

(ii) Ordinea de oblicitate a acestor constituenți nominali față de centrul lor este captată prin trăsătura ORD-OBLIC- VAL, despre care s-a vorbit mai sus.

(iii) Din această ordine se abstrage acum conceptul de c o m a n d ă o b l i c ă ("comandă datorată oblicității"). Comanda oblică este un concept relațional. El definește o relație între constituenții înscriși pe lista de oblicitate. Relația spune că acel constituent care este cel mai puțin oblic comandă oblic oricare alt constituent mai oblic decât el. O consecință imediată este că, dacă subiectul nominal există, el va comanda oblic toți constituenții din listă.

(iv) Anaforicitatea unui constituent înseamnă dependența sa referențială de un alt constituent din același enunț. Această dependență se indică prin intermediul coindexării.

Din aceste patru elemente se poate acum construi o caracterizare a constituenților susceptibili sau nu de comportament anaforic. Aceste

caracterizări se concentrează în principiile dependenței anaforice.

Principiul A: Un constituent marcat anaf se coindexează cu cel mai puțin oblic constituent care îl comandă oblic.

Principiul B: Un constituent marcat nonanaf nu se coindexează cu nici un alt constituent care îl comandă oblic.

Principiul C: Un constituent marcat ob-nonpron nu se coindexează cu nimeni.

Sub incidența Principiului A cad reflexivele obiective, dinamice, eventive și reciproce. Constituentul cel mai puțin oblic ce le comandă este subiectul, de unde și coindexările (pe care le dăm într-o manieră informală):

(11) |1|*Ion* |1|*se consolează că va câștiga altădată.*

(12) |2|*Ion* |2|*se teme de represalii.*

(13) |3|*Ion* |3|*s-a albit când l-a văzut pe Vasile.*

(14) |4|<*Ion și Ioana*> nu |4|*se mai suportă.*

Domeniul principiului B este cel al pronumelor personale. Principiul C reglementează comportamentul substantivelor.

Principiile dependenței anaforice prezentate de Pollard și Sag diferă de cele cunoscute din Chomsky 1981 prin aceea că sunt derivate din dependențe lexicale, și nu din configurații sintactice.

5.3.6. Principiul Trăsăturilor Nonlocale (PTN)

Regula lexicală care permite trecerea de la M7 la M8 pune în evidență existența unui tip verbal, denumit mai sus tranz-el-verb. Caracteristica lui este faptul că deține o trăsătură pe care "corespondentul" său n-o conține --trăsătura

NONLOC. Valoarea acestei trăsături pentru structurile de tipul (5), reluat aici ca:

(15) *Tablouri ca acestea, Muzeul Național nu va achiziționa niciodată,* este obiectul de tipul abs.

Această regulă lexicală deschide calea de analiză a structurilor de tipul (15) --și nu numai acestora --, însă nu reprezintă și o soluție efectivă pentru dependența la distanță, pe care structurile de acest tip o expun. O problemă rămâne în suspensie: cum se poate face legătura cu faptul că un constituent absent ajunge să fie găsit într-o altă parte a enunțului?

Pentru a răspunde la această întrebare, HPSG folosește o regulă sintagmatică deja amintită în paragrafele precedente: tipul c-antep. Este acum evident că un verb de tipul tranz-el-verb (sau de un tip încă și mai general, cum ar fi el-verb, adică un tip în care se specifică doar că una dintre valențele verbului e satisfăcută prin elipsă) se înserează într-o structură de tipul c-antep.

Detaliile acestui tip sintagmatic nu au fost până acum descrise, însă, la acest punct, amănuntele devin necesare.

M12

$$c - antep \left[\begin{array}{l} \text{SINSEM} \mid \text{LOC} \mid \text{CAT} : [\text{CENTRU} : |1|] \\ \text{RAM} : \left[\begin{array}{l} \text{R} - \text{C} : \text{ram} - c[\text{SINSEM} \mid \text{LOC} \mid \text{CENTRU} : |1|] \\ \text{R} - \text{ANTEP} : \text{ram} - antep[\text{CENTRU} : \text{nume}] \end{array} \right] \end{array} \right]$$

M12 formulează în termeni matriciali ceea ce găsim în următoarea regulă de rescriere, cunoscută în gramaticile generative drept "regula de adjuncție a lui Chomsky" (este însă de precizat că, în HPSG, echivalentul de notație al acestei reguli nu e folosit pentru a da seama de fenomenul denumit de Chomsky *left dislocation* "dislocare la stânga". În HPSG dislocările la stânga sunt tratate

drept structuri sintagmatice de tipul "centru-adjunct"):

RS(3) S \rightarrow N, S

Din M12 lipsește însă ceva esențial, și anume precizarea că valoarea trăsăturii CENTRU în ram-antep trebuie să fie aceeași cu valoarea trăsăturii ABS din M8.

S-ar părea că nimic nu ne împiedică să apelăm la procedura coindexării pentru a umple această lacună. Această soluție nu trebuie totuși aleasă cu atâta ușurință dintr-un motiv simplu: coindexarea este doar un mecanism, nu un principiu. Aplicațiile ei se cer a fi reglementate, iar reglementările ei se cer a fi explicite. Tocmai de o asemenea reglementare e nevoie în contextul de față; o reglementare care să corespundă intuiției vorbitorului că acel constituent-lipsă al verbului *a achiziționa* este același cu constituentul antepus.

Reglementarea în cauză este adusă de către Principiul Trăsăturilor Nonlocale. Acest principiu arată cum se propagă o trăsătură precum ABS din M8. Principiul are forma următoare:

PTN Orice trăsătură nonlocală se propagă de pe (sub)constituentul ram-x (cu x variabilă pentru compl, sub) pe constituentul căruia ram-x îi aparține.

PTN precizează așadar că o trăsătură nonlocală 'migrează' de la constituentul lexical care o are în proprietate spre constituenții de ordin succesiv superior în care constituentul lexical se inserează. Ea se va propaga până la punctul la care va întâlni o barieră. Iar bariera este exact tipul sintagmatic c-antep. Acest tip sintagmatic introduce constituentul a cărui valoare pentru trăsătura locală este aceeași cu valoarea trăsăturii NONLOC. Abia prin această precizare

coindexarea dintre valoarea trăsăturii LOC a constituentului antepus și valoarea trăsăturii NONLOC a ramurii sale vecine este reglementată. Este ceea ce arată matricea M13:

M13

$$c - antep \left[\begin{array}{l} \text{SINSEM} \mid \text{LOC} \mid \text{CAT} : [\text{CENTRU} : | 2 |] \\ \text{RAM} : \left[\begin{array}{l} \text{R} - \text{C} : \text{ram} - c \left[\text{SINSEM} : \left[\begin{array}{l} \text{LOC} \mid \text{CENTRU} : | 2 | \\ \text{NONLOC} : \text{abs}[\text{ABS} : \{ | 1 | \}] \end{array} \right] \right] \\ \text{R} - \text{ANTEP} : \text{ram} - \text{antep} [\text{SINSEM} \mid \text{LOC} \mid \text{CENTRU} : | 1 | \text{ nume}] \end{array} \right] \end{array} \right]$$

Studiul trăsăturilor nonlocale este unul dintre cele mai interesante în sintaxa unei limbi. În această privință, HPSG oferă numai cadrul de analiză și principiile ei. În rest, totul rămâne de făcut, deoarece nu este înscris în natura limbilor că peste tot se găsesc aceleași trăsături nonlocale sau că exact aceiași constituenți sunt supuși 'dislocării' .

6. REGULI DE PRECEDENȚA LINIARA (RPL)

Ordinea cuvintelor este tratată în HPSG într-o componentă separată. Această componentă cuprinde judecăți precum:

RPL Constituentul *x* precedă constituentul *y*.

Dincolo de astfel de judecăți, HPSG se străduiește totuși să formuleze judecăți de o mai mare generalitate. O asemenea judecată e, de exemplu, următoarea:

RPL1 Dacă *x* este un constituent lexical, iar *y* complementul său,

atunci x precedă y .

RPL2 Complementele nominale se succedă în ordinea oblicității lor crescânde.

Astfel de generalizări sunt în mod precis valabile pentru limba engleză, dar rămâne de verificat dacă sunt utile și pentru alte limbi.

Un tratament sistematic al problemelor de topică ar putea rezulta din examinarea constituenților de o complexitate succesivă. Pe această cale, ar fi de studiat mai întâi ordinea în structuri cu constituent lexical și clitice, apoi în structuri cu constituent lexical și valențe lexicale ș.a.m.d. Această cale de analiză ar putea arăta în ce măsură un tratament 'compozițional' al topicii ar fi posibil și eficace.

7. CONCLUZII

HPSG este un curent aflat într-o mișcare ce-i probează vitalitatea. Teoria nu este definitivată, dar acest lucru nu trebuie să îndemne la expectativă, căci se poate foarte bine ca găsirea unei 'formule de stabilitate' să coincidă cu istoricizarea curentului.

Aplicațiile HPSG sunt deja numeroase și nu se mai limitează doar la domeniul limbii engleze. Substanțiale sunt, de pildă, analizele efectuate pe limbile romanice, în special pe franceză și italiană (Monachesi 1995).

BIBLIOGRAFIE

- Anne Abeillé, *Les nouvelles syntaxes. Grammaires d'unification et analyse du français*, Armand Colin 1993.
- Anne Abeillé și Daniele Godard, *La complémentation des auxiliaires français*, în "Languages", 122, 1996, 32-61
- John Barwise și John Perry, *Situations and Attitude*, The MIT Press, 1983
- Joan Bresnan (ed.), *The Minimalist representation of Grammatical Relations*, The MIT Press, 1982
- Noam Chomsky, *Lectures on Government and Binding*, Foris, Dordrecht, 1981
- idem, *Bare Phrase Structure*, în Webelhuth (ed.) 1995, 383-439
- Gerald Gazdar, Ewan Klein, Geoffrey Pullum și Ivan Sag, *Generalized Phrase Structure Grammar*, Basil Blackwell, 1985
- Paola Monachesi, *A Grammar of Italian Clitics*, ITK. Dissertation Series, Tilburg, 1995
- Richard Oehrle, Emmon Bach și Deirdre Wheeler (eds.), *Categorial Grammar and Natural Language Structures*, Dordrecht, Reidel, 1988.
- Carl Pollard și Ivan Sag, *Information-based Syntax and Semantics. Fundamentals*, vol.1, CSLI, Stanford, 1987 .
- Idem, *Head-driven Phrase Structure Grammar*, University of Chicago Press and CSLI, Stanford, 1994.
- Ivan Sag, Gerald Gazdar, Thomas Wasow și Steven Weisler, *Coordination and How to Distinguish Categories*, în *Natural Language and Linguistic Theory*, 1988, 117-169

Stuart Shieber, *An Introduction to Unification-based Approaches to Grammar*, CSLI, Stanford, 1986 Gert Webelhuth (ed.), *Government and Binding and the Mentalist Program*, Basil Blackwell, 1995.

**ARGUMENTE, VALENȚE ȘI DEPENDENȚI.
ANTICIPAREA COMPLEMENTULUI DIRECT ÎN LIMBA ROMÂNĂ
DIN PERSPECTIVA HPSG**

Verginica Barbu și Emil Ionescu

1. Introducere

1.1. Cadrul teoretic

În ultimii cinci ani, mai multe lucrări HPSG au prezentat dovezi numeroase că nu se poate stabili în toate cazurile o distincție clară între adjuncti și complemente¹ (în acest sens, vezi datele citate de Bouma, Malouf și Sag (1999) p. 37-39).

Pe de altă parte, dezvoltări teoretice recente în HPSG fac distincția între valențe (VAL), argumente (ARG-ST) și dependenți (DEPS, a se vedea Bouma, Malouf și Sag (1999)). Atributul VAL face legătura între cadrul de subcategorizare al intrării lexicale și proiecțiile sintagmatice corespunzătoare; ARG-ST este folosit în fenomenele de legare, în timp ce trăsătura DEPS este introdusă pentru un tratament uniform al extracției valențelor și adjunctilor.

Analiza pe care o propunem aici - având drept obiect anticiparea complementului direct în limba română (ACD) - se bazează pe ideea ambiguității adjunct-complement și pe distincția valențe-argumente-dependenți.

1.2. ACD în română. Scurtă prezentare a analizelor sale generative

Asemenea unor dialecte spaniole și câtorva limbi balcanice, limba română prezintă o structură de „anticipare clitică” a complementului direct. De exemplu:

(1) Nu-l_i cunosc pe Ion_i

Exemplele de tipul (1) reprezintă o provocare pentru orice teorie gramaticală care asumă că un argument are o singură realizare. Astfel, (1) este în aparență un exemplu

¹ Termenii “adjunct” și “complement” sunt folosiți aici în sens nonconfigurațional: ei denumesc “funcții”, nu “poziții”.

care nu respectă această constrângere, deoarece obiectul direct al verbului **a cunoaște** pare să fie realizat atât ca un pronume neaccentuat, cât și ca un grup nominal (GN)². La prima vedere așadar, fie (1) trebuie declarat negramatical, fie constrângerea asupra realizării argumentului trebuie abandonată (sau modificată). Se știe însă că nici una din aceste soluții nu este satisfăcătoare și, de fapt, strategia obișnuită este de a arăta că anticiparea clitică este doar un aparent contra-exemplu la regula că un argument are o singură realizare. Aceasta va fi și strategia pe care o vom adopta aici.

Există deja o literatură bogată dedicată ACD în limba română. Cele mai recente contribuții vin din teoria GB. Principalele teze/ipoteze ale acestei abordări sunt următoarele:

Poziția sintactică în care apare cliticul

Ipoteza 1

Cliticul este generat în aceeași poziție ca orice GN-complement al verbului. El se deplasează în IP (= GFlex), care este o proiecție lipsită de specificator (Dobrovie-Sorin (1994) p. 54).

Ipoteza 2

Cliticul este generat ca parte a verbului. La rândul său, verbul este un complex lexical (a se vedea Borer (1984), pentru clitice în general, și Cornilescu (1987), pentru cliticele pronominale în română).

Statutul cliticului

Cliticul poartă informație cazuală. Totuși, de vreme ce nu apare într-o poziție structurală, el nu poate fi argument al verbului. Verbul atribuie rol tematic doar GN (a se vedea Jaeggli (1986) p. 17-18, pentru clitice în general, și Cornilescu (1987) p. 214, pentru clitice în română).

Secvența $cl_i \dots GN_i$

Cliticul și GN pe care cliticul îl anticipează formează un constituent discontinuu (Cornilescu (1987) p. 215). Cliticul este centrul, iar GN ocupă poziția de specificator (Gierling (1998) p. 79).

² În această lucrare, termenii “pronume neaccentuat” și “clitic” vor fi considerați sinonimi.

Ipoteza 1

GN anticipat primește caz de la prepoziția **pe** (Cornilescu (1987), Dobrovie-Sorin (1994)).

Ipoteza 2

GN anticipat de clitic primește caz prin acord, într-o configurație specificator-centru (Gierling (1998) p.79).

Semantica

GN anticipat nu are proprietăți cuantificaționale. El este referențial (Dobrovie-Sorin (1994) p. 234-235).

Sînt multe observații pătrunzătoare în aceste analize, dar nu toate pot fi acceptate. De exemplu, nu este clar cum se poate împăca afirmația că cliticul este nonargumental cu comportamentul său în structuri fără anticipare, în care în mod evident el este un argument al verbului (vezi mai jos, secțiunea 4.1.1.). Altă problemă este următoarea: dacă GN dublat este deplasat în poziția de specificator în GClitic (după cum asumă Gierling), cât de sus trebuie să urce în continuare cliticul însuși pentru a lăsa în urmă GN și pentru a justifica astfel ordinea cl ... GN; ? Și, în sfârșit, care sunt motivele pentru a considera că **pe** este o prepoziție?

Propunem în continuare o analiză HPSG a fenomenului, care evită aceste aspecte problematice. Ea poate fi rezumată astfel:

- Într-o structură de ACD, verbul încorporează cliticul ca parte a morfologiei sale extinse.
- Într-o structură de ACD, cliticul nu este nici argument, nici valență a verbului; el este doar *dependent*. Cazul acuzativ este cel care indică acest statul al cliticului: cazul cliticului este atribuit de către verb.
- Într-o structură de ACD, GN anticipat are un comportament bivalent: el are proprietăți de complement în relația sa cu verbul și de adjunct în relația cu cliticul.

În secțiunea 2 descriem principalele trăsături ale fenomenului de ACD în română. În secțiunea 3 identificăm elementele care duc la o analiză corespunzătoare. Secțiunea 4

este dedicată unei analize a GN anticipat, iar secțiune 5 prezintă fenomenul ACD din perspectiva conceptelor HPSG menționate în introducere.

2. ACD în română: descriere

ACD în română are următoarele proprietăți³:

1. Verbul (tranzitiv) este însoțit de un pronume (personal) neaccentuat din paradigma de mai jos:

	SINGULAR	PLURAL
PERSOANA ÎNTÂI	mă, m-	ne
PERSOANA A DOUA	te	vă, v-
PERSOANA A TREIA	îl, îl(-) (masc.); o (fem.)	îi, îi(-) (masc.); le (fem.)

2. GN anticipat prin clitic apare în poziție post-verbală și este obligatoriu precedat de așa-numita prepoziție *pe*⁴:

(2) Nu-l_i cunosc *(pe) el_i

Observație

Spre deosebire de italiană, în română subiectul și verbul nu sunt în mod obligatoriu adiacente. Astfel, structurile din română în care subiectul urmează verbului și este urmat de un obiect direct dublat prin clitic sunt structuri de ACD (a se vedea (3)). În italiană însă (a se vedea (4)) ele sunt *dislocări la dreapta cu clitic*.

(3) O iubește Ion *pe Ioana*.

(4) Lo legge Gianni, *il giornale*.⁵

AC cl-îl_i citește Ion, ziarul_i
„Citește Ion ziarul.”

3. ACD constă în faptul că pronumele neaccentuat și GN dublat se acordă în

³ Plasamentul și liniarizarea cliticelor nu fac obiectul acestei lucrări (vezi Dobrovie-Sorin (1994) p. 49-81, pentru o analiză din perspectivă GB, și (Monachesi (2000) pentru o analiză din perspectivă HPSG).

⁴ Dacă grupul nominal este în poziție preverbală avem de-a face cu o structură de dislocare la stânga cu clitic. De exemplu: (i) Eu romanul_i l_i-am citit.

⁵ Exemplul este împrumutat din Sanfilippo (1997) p. 357.

caz, persoană, număr și (pentru persoana a treia) în gen.

4. Clasa de GN anticipate este o subclasă a setului de GN în acuzativ. De exemplu, formele accentuate ale pronumelui personal în acuzativ trebuie să fie anticipate:

(8) Nu-*(I₁) cunosc **pe el** ,

În schimb, cuantificatorul **cineva** nu poate fi anticipat:

(6) Ion (* îl/o) urăște **pe cineva**.

5. Formele tari ale pronumelui anticipat prin clitic cad obligatoriu sub incidența focusului. Forma nemarcată prin focus conține doar pronumele neaccentuat:

(7) (a) Ion **mă** vizitează.

Versiunea cu focus a lui (7)(a) este o structură ACD. Acest focus *in situ* este exprimat prin intonație, prin determinatori corespunzători (și anume, **chiar**, **numai**, **doar**, **și**) sau prin ambele:

(7) (b) Ion mă vizitează (numai/chiar/doar/și) **PE MINE**.

GN nepronominale anticipate se marchează opțional prin focus (a se vedea (8)). Dacă însă se încearcă să *nu se marcheze* prin focus un GN *pronominal*, ceea ce se obține este o structură cel puțin ciudată (vezi (9)):

(8) Ion îi așteaptă pe părinții săi/PE PĂRINȚII SĂI. (9) Ion te așteaptă pe tine (??) / PE TINE.

3. Abordarea problemei

Pentru a da o explicație caracteristicilor enumerate la punctele 1–4, trebuie să răspundem la următoarele întrebări⁶:

- (i) Care este statutul cliticelor de acuzativ?
- (ii) Care este statutul lui **pe**?
- (iii) Care este statutul GN anticipat?

⁶ Proprietatea de la punctul 5 nu este implicată în nici un fel în construcția argumentației din această lucrare.

Pentru a răspunde la primele două întrebări, vom prelua concluziile câtorva analize anterioare (Ionescu (1996), Barbu (1998), Monachesi (2000), Ionescu (2001)). Acestea sunt următoarele:

Cliticele de acuzativ

- (i) Cliticele de acuzativ nu trec testele de constituență, precum coordonarea și interogativizarea. În consecință, nu pot fi considerate constituenți lexicali și nu pot avea independență sintactică.
- (ii) În schimb, ele răspund testelor pentru afixe. Astfel, ele manifestă un grad înalt de selecție a 'gazdei lexicale'; ele pun în evidență neregularități în seturile de combinații cu această gazdă; au o topică fixă și foarte specifică; în sfârșit, nu au drept 'rază de acțiune' coordonarea (Sag and Miller (1997)). Pentru toate aceste motive vom considera cliticele de acuzativ drept parte a verbului tranzitiv⁷.

„Prepoziția” pe

- (i) **Pe** din construcțiile cu obiect direct nu este nici prepoziție, nici element care atribuie caz.
- (ii) **Pe** marchează un GN în acuzativ dar nu orice GN în acuzativ este marcat de **pe**.
- (iii) **Pe** este un „marcător de oblicitate”.
- (iv) Sintagma **pe** care **pe** o formează cu grupul nominal-centru este un subtip (specific limbii române) al tipului sintagmatic centru-marcător (*hd-mark-ph*) (pentru *hd-mark-ph* vezi Pollard și Sag (1994) p. 44-46). Vom abrevia acest subtip *hd-OBL mark-ph* (a se vedea Ionescu (2001))
- (v) **Pe** are un conținut semantic vid (fapt care nu trebuie pus în legătură cu acela că **pe** marchează doar unele tipuri semantice de GN în acuzativ).

4. Statutul GN anticipat

În această secțiune vom arăta că GN anticipate au proprietăți specifice complementelor (dacă sunt raportate la verb) și proprietăți de adjuncți (în relația lor cu perechea clitică). Iată câteva trăsături necontrovertate ale complementelor:

⁷ Această analiză se întâlnește cu interpretarea dată de Valeria Guțu-Romalo verbelor auxiliare, pe care le consideră elemente afixale și nu lexicale (Guțu-Romalo (1968)). Analiza noastră este de asemenea compatibilă cu aceea a Anei-Maria Barbu, care demonstrează că alături de pronumele neaccentuate și de verbele auxiliare, semiadverbele și negația verbală au același statut afixal (Barbu (1998)).

- sunt argumente ale centrului;
- apar în lista de valențe ale acestuia;
- sunt argumente semantice;
- ocurența lor este, în general, obligatorie.

Pe de altă parte, adjuncții sunt funcționari semantici în relație cu centrul lor sintactic și, în general, nu sunt valențe ale centrului pe care îl modifică.

4.1. Proprietăți specifice complementului manifestate de GN anticipat

Dovezi că GN dublate prezintă proprietăți specifice complementelor vin din fenomenul de legare, din atribuirea rolului semantic și din pasivizare.

4.1.1. Atribuirea rolului semantic și legarea

GN anticipate care sunt nepronominale arată în mod clar că primesc rol semantic de la verb și că sunt implicate în relații de legare. Acest lucru pledează pentru statutul lor argumental. În ceea ce privește fenomenul de legare, dacă un GN anticipat este o expresie referențială, el va fi liber, după cum arată exemplul (10)⁸:

(10) Ion_i crede că poliția_j îl urmărește pe Petre_{i/*j/k}

Trebuie remarcat aici comportamentul neuniform al cliticelor de acuzativ din punctul de vedere al legării. Dacă un pronume neaccentuat nu anticipează un GN, el se comportă conform statutului său pronominal, adică este liber în categoria sa de guvernare:

(11) Ion_i crede că poliția_j îl_{i/*j/k} urmărește.

Dimpotrivă, dacă pronumele neaccentuat dublează o expresie referențială, cea care contează pentru legare este expresia referențială și nu cliticul. De exemplu, în (10), rescris mai jos ca (12), pronumele neaccentuat *îl* nu poate fi legat de GN *Ion*, deoarece GN *pe Petre* este o expresie referențială:

(12) Ion_i crede că poliția_j îl_{i/*j/k} urmărește *pe* Petre_{i/*j/k}

Aceasta este o dovadă că cliticele de acuzativ *nu sunt argumente*.

⁸ Pentru discuția de față, nu este relevant la ce teorie a legării aderăm.

4.1.2. Pasivizarea

Indiferent la ce teorie a pasivizării aderăm, este aproape general admis că pasivizarea afectează argumentul obiect direct al verbului⁹. Dacă ne bazăm pe această premisă minimală, ajungem la concluzia că un GN dublat este un complement al verbului, așadar un argument, deoarece într-o structură pasivă subiectul este GN dublat din structura activă:

(13) Poliția îl urmărește *pe* Ion.

(14) Ion este urmărit de către poliție.

4.2. Proprietăți specifice adjuncților

Relevante pentru natura de adjunct a GN dublat sunt reacția la testul pronominalizării, felul în care GN anticipat primește caz și relația semantică dintre GN anticipat și copia sa clitică.

4.2.1. Pronominalizarea

În română, obiectul direct trece testul pronominalizării, permițând substituirea sa sistematică cu cliticul de acuzativ corespunzător. În această situație, un pronume neaccentuat devine argument al verbului:

(15) (a) Ion citește **o carte**. (b) Ion **o** citește.

Pronominalizarea nu mai e însă posibilă în cazul GN dublat. Dacă un astfel de GN este substituit cu un clitic de acuzativ, construcția devine incorectă:

(16) (a) Ion **o** iubește pe Ioana. (b) * Ion **o o** iubește.

Incorectitudinea nu se explică aici prin faptul că pronumele neaccentuat care înlocuiește GN dublat este un argument redundant al verbului (îndeplinind, adică, o funcție deja îndeplinită de alt pronume neaccentuat). Într-adevăr, după cum arată legarea, cliticul *nu este* un argument. Doar GN dublat are proprietăți de argument.

În sine, imposibilitatea de a pronominaliza un GN anticipat nu oferă dovezi că acesta are proprietăți de adjunct. Acest test arată doar că GN dublate nu sunt obiecte

⁹ Există, bineînțeles, și verbe tranzitive care nu admit pasivizarea (**a comporta**, **a avea** etc). Această subclasă de verbe nu reprezintă însă un contra-exemplu la argumentul invocat aici. Noi ne construim argumentul observând ce se întâmplă cu obiectul direct *atunci când verbul admite pasivizarea*, și nu observând *dacă verbul admite sau nu pasivizarea*.

directe obișnuite – deoarece obiectele directe obișnuite participă la legare și pot fi pronominalizate în română. Cu toate acestea, ceea ce face ca imposibilitatea pronominalizării să fie relevantă este faptul că *întreaga secvență clitic-GN poate fi înlocuită cu un clitic*:

(17) Ion o iubește **pe Ioana**.

(18) Ion o iubește.

Înlocuirea secvenței discontinue **o...pe Ioana** cu pronumele neaccentuat **o** în (17) nu este o pronominalizare "în gol"¹⁰. Coroborată cu imposibilitatea pronominalizării în contexte ca (16) (b), substituția din (17) duce la următoarele concluzii:

- (a) Secvența clitic-GN funcționează ca un fel de 'sintagmă' în care cliticul pare să fie centrul, iar GN pare să fie un constituent noncentral.
- (b) De vreme ce cliticul este ca parte de vorbire un nume, întreaga structură clitic-GN este un GN.

Dacă aceste concluzii sunt adevărate, secvența clitic-GN se aseamănă fie unei sintagme de tipul *centru-specificator*, fie uneia de tipul *centru-adjunct*. În următoarele două secțiuni, vom arăta de ce alte tipuri de sintagme (centru-complement sau centru-subiect) sunt excluse din investigație.

4.2.2. Marcajul cazual

Un loc comun în literatura dedicată ACD în română este acela că GN dublat primește cazul acuzativ de la prepoziția **pe** care îl precedă. Nu există însă dovezi convingătoare că în structurile de ACD, **pe** este o prepoziție. Dimpotrivă, există motive întemeiate pentru a afirma că acest **pe** este un marcator al cărui rol este de a specifica un GN¹¹. În această ipoteză, GN marcat primește caz în mod independent.

Plecând de la ipoteza că **pe** este un marcator neimplicat în atribuirea de caz, nu se poate afirma că într-o structură ACD, *verbul* este cel care atribuie cazul acuzativ GN anticipat. Verbul atribuie caz doar cliticului, ceea ce înseamnă că este imposibil ca el să atribuie caz și GN anticipat. Ceea ce este important aici este *acordul în caz între clitic și GN anticipat*. Acest acord indică în mod decisiv că GN anticipat *nu*

¹⁰ Pronominalizarea "în gol" ar însemna, în acest caz, substituția, în secvența **o...pe Ioana**, a unei ocurențe a lui **o** cu o altă ocurență a lui **o**.

¹¹ Argumente numeroase în acest sens se găsesc în Ionescu (2001)

primește acuzativ prin selecția argumentală exercitată de către verb. Cazul GN dublat este așadar dobândit prin relația de acord pe care o are cu copia sa clitică.

Corelând aceste date cu cele oferite de pronominalizare, ajungem la noi dovezi că o astfel de secvență exemplifică fie tipul sintagmatic centru-specificator, fie pe cel de centru-adjunct: într-adevăr, doar în astfel de structuri constituentul care nu este centru primește caz prin acord de la centrul său. Cu toate acestea, încă nu avem dovezi pentru a preciza care dintre aceste două tipuri de sintagme este cea implicată în structurile de ACD.

4.2.3. Relația semantică dintre GN dublat și clitice

O particularitate bine cunoscută a adjuncților este calitatea lor de functori semantici: ei sunt centrii semantici ai sintagmelor cărora le aparțin., deoarece ei modifică semantic centrul sintactic. Această proprietate a adjuncților este ilustrată de felul în care un GN anticipat se comportă față de cliticul său pereche. După cum se știe, un clitic nu are conținut descriptiv. Cu toate acestea, faptul că el are aceeași informație de persoană, număr și gen ca și un GN anticipat, îi permite acestuia din urmă să modifice conținutul vid al cliticului. Relația este una *de modificare, nu de specificare*, deoarece contribuția GN anticipat la conținutul GN ‘proiectat’ de clitic nu este vidă (ca în cazul specificatorilor): din perspectiva relației sale cu cliticul, GN dublat deține o informație semantică nouă, adică o informație care nu este prezentă în conținutul cliticului.

Acest comportament de adjunct al GN anticipat devine și mai evident dacă este privit din perspectiva semanticii formale. Într-o lucrare despre limbile care permit ACD, Guttiérrez-Rexach (Gutiérrez-Rexach (1999)) dovedește că cliticele de acuzativ denotă ceea ce într-o teorie a cuantificatorilor generalizați se numește *filtru principal*. Deși autorul nu insistă asupra denotației întregii secvențe clitic-GN, este destul de clar că și aceasta este tot un filtru principal. Dar cum am putea atunci interpreta denotația GN anticipat? Un răspuns plauzibil este acela că denotația GN dublat este o funcție de la un filtru principal (cliticul) la alt filtru principal (secvența clitic-GN). Însă a afirma acest lucru înseamnă a recurge la modelul modificării nominale care tocmai a fost invocat mai sus.

5. Analiza ACD în HPSG

5.1. Adjuncții-complement

Dovezile de până acum arată că proprietățile specifice adjuncților și cele specifice complementelor manifestate de GN anticipate sunt la fel de importante. De fapt, această combinație de trăsături definește natura acestor constituenți. Ar fi o greșeală să se afirme că un set de proprietăți este mai important decât celălalt.

Natura duală a GN anticipate în română reprezintă o nouă dovadă pentru ceea ce s-ar putea numi „teoria adjuncților- complement”¹². După cum am menționat în introducerea acestei lucrări, această teorie este specifică multor lucrări HPSG și se bazează în primul rând pe dovezi că un adjunct ar putea fi considerat și complement. Faptele comentate mai sus vor fi în continuare analizate din această perspectivă.

Pentru o astfel de analiză, sunt necesare două etape. Mai întâi trebuie să impunem o constrângere prin care GN anticipat să modifice cliticul. Apoi va trebui să precizăm forma intrărilor lexicale care dau seama de principalele tipuri de construcții tranzitive din limba română¹³.

5.1.1. Constrângeri asupra GN anticipat

Constrângerea necesară asupra GN anticipat trebuie să conțină informația că el modifică cliticul. De vreme ce GN anticipat e marcat de **pe**, constrângerea trebuie impusă asupra tipului sintagmatic centru-marcator de oblicitate (*hd-OBL marker-ph*). Pe de altă parte însă, de vreme ce nu orice sintagmă de acest tip modifică un verb cu clitic, constrângerea trebuie să fie prin excepție:

C1¹⁴:

$$/hd-OBLmark-ph \Rightarrow \left[HEAD:noun \left[MOD:aff-ss \left[\begin{array}{l} HEAD:noun \{ CASE:acc \} \\ CONT|IND:ref \end{array} \right] \right] \right]$$

(În C1, / este simbolul pentru excepție, iar \Rightarrow înseamnă implicație.) C1 trebuie citit astfel: „dacă nu intervine o altă constrângere mai specifică, o sintagmă de tipul *centru-marcator de oblicitate* trebuie să modifice informația sintactico-semantică (= *synsem*-ul, în terminologia HPSG) a afixului indicat ca valoare a trăsăturii MOD(IFICĂ)”.

C1 este urmat de patru constrângeri care descriu acele sintagme *centru-marcator* ce *nu* modifică clitice: acestea sunt GN marcate de **pe**, care nu apar niciodată în structuri de ACD. Cu alte cuvinte, aceste GN sunt obiecte directe selectate de verbe fără clitic încorporat.

¹² Cu singura diferență că în cazul nostru GN anticipat manifestă proprietăți de complement în relația cu verbul care-l guvernează și proprietăți de adjunct în relație cu cliticul.

¹³ În ciuda unor explicații, această parte a lucrării presupune o anumită familiarizare a cititorului cu HPSG.

¹⁴ Din considerente de uniformitate, preferăm să păstrăm notația standard, adică aceea din limba engleză.

Specificarea [MOD: *ev*] înseamnă că GN nu modifică nimic. Această specificare apare ca parte a reprezentărilor lexicale ale numelor cuprinse în acele sintagme de tipul *centru-marcator de oblicitate* ce sunt excepții la C1. Specificarea este transmisă sintagmei înseși prin Principiul Trăsăturilor Centrale. Numele în cauză sunt **cineva**, **oricine**, **cine**, **nimeni**. Specificarea [MOD: *ev*], deținută de fiecare dintre ele, explică contrastul de mai jos:

(19) (a) Ion a întâlnit **pe cineva**. (b) * Ion l-a întâlnit **pe cineva**.

(20) (a) Ion primește **pe oricine** în casa lui. (b) * Ion îl primește **pe oricine** în casa lui.

(21) (a) Ion nu a învinovățit **pe nimeni**. (b) * Ion nu l-a învinovățit **pe nimeni**.

(22) (a) **Pe cine** nu iubește Ion ? (b) * **Pe cine** nu-l iubește Ion ?

În exemplele (b), nongramaticalitatea provine din faptul că un GN marcat de **pe** și specificat ca [MOD: *ev*] este complementul unui verb care cere un grup nominal marcat de **pe**, și a cărui valoare pentru trăsătura MOD trebuie să fie cea din C1 (și, prin urmare, distinctă de valoarea vidă – *e*(mpty) *v*(alue)).

5.1.2. Reprezentările lexicale ale tranzitivității verbale

Analiza de mai sus arată că în română construcțiile tranzitive își au originea ultimă în trei tipuri de intrări lexicale:

(i) Un verb tranzitiv poate selecta ca obiect direct un GN:

(23) Ion iubește **femeile**.

(ii) Un verb tranzitiv poate avea ca argument un clitic:

(24) Ion l-a întâlnit.

(iii) Un verb tranzitiv poate selecta un obiect direct anticipat de un clitic care nu este argument¹⁵:

(25) Ion a ignorat-**o** **pe Ioana**.

Este posibil ca “același verb” să satisfacă toate aceste tipare. De exemplu, verbul **a iubi** poate avea drept obiect direct un GN (Ion iubește **femeile**), un argument clitic (Ion **le**

¹⁵ Aceste tipuri ne-au fost sugerate de Ana-Maria Barbu.

iubește) sau un GN anticipat (Ion le iubește pe toate colegile de serviciu., Ion le iubește **ȘI PE ELE.**)

Diferențele între aceste tipuri de verbe tranzitive pot fi ușor exprimate folosind trăsăturile VAL, ARG-ST și DEPS (vezi Bouma, Malouf and Sag (1999)).

Valorile acestor trăsături sunt liste de *synsem*-uri. Relația dintre ele poate fi redată astfel:

- Nu orice valență este un argument (pot exista subiecte și obiecte expletive) și de asemenea nu orice argument este o valență (un argument poate avea un *synsem* canonic, în timp ce o valență are întotdeauna unul necanonic¹⁶).
- Orice valență este un dependent, dar reciproca nu este în mod necesar adevărată (deoarece un dependent poate avea un *synsem* noncanonic, în timp ce *synsem*-ul unei valențe este întotdeauna canonic).
- Orice argument este un dependent, dar reciproca nu este în mod necesar adevărată (deoarece este posibil ca un dependent să nu joace nici un rol în structura argumentală).

Cu aceste specificări, informațiile relevante ale fiecăruia dintre tipurile lexicale identificate anterior arată astfel:

(i') Un verb tranzitiv care nu conține clitic trebuie să satisfacă condițiile descrise în matricea M1:

M1

$$\left[\begin{array}{l} \text{HEAD : trans - verb} \\ \text{VAL : } \left[\begin{array}{l} \text{SUBJ : } |1| \\ \text{COMPS : } |2| \text{ss} [\text{HD : noun} [\text{MOD : ev}]] \end{array} \right] \\ \text{ARG - ST : } |1| \oplus |2| \\ \text{DEPS : } |1| \oplus |2| \end{array} \right]$$

După cum se poate vedea în M1, un astfel de verb selectează drept

¹⁶ Un *synsem* canonic este *synsem*-ul unui obiect lingvistic care poate participa în structuri sintactice. De exemplu, *synsem*-ul unui cuvânt sau al unei sintagme este canonic. Un *synsem* noncanonic, pe de altă parte, este *synsem*-ul unui obiect lingvistic care nu poate fi implicat în construcții sintactice. *Synsem*-urile afixelor sau ale "categoriilor vide" sunt noncanonice.

complement un GN care trebuie să fie specificat că nu modifică nimic ([MOD: *ev*]). Ceea ce este o valență (adică un subiect sau un complement) este și un argument, și ceea ce este un argument este și un dependent. Complementul este nespecificat în raport cu marcarea cu **pe**, ceea ce e corect, deoarece complementul unui verb tranzitiv “nonclitic” poate fi sau marcat cu **pe**, sau nemarcat.

(ii’) Un verb care încorporează un clitic fără funcție de anticipare trebuie să respecte constrângerea din M2 (numim verbele cu clitic încorporat “verbe tranzitiv-clitice” (*trans-cl-verb*))¹⁷:

M2

$$\left[\begin{array}{l} \text{HEAD : } trans - cl - verb \\ \text{VAL : } \left[\begin{array}{l} \text{SUBJ : } |1| \\ \text{COMPS : } el \end{array} \right] \\ \text{ARG - ST : } |1| \oplus |2| \text{aff} - ss \left[\begin{array}{l} \text{HEAD : } noun[\text{CASE : } acc] \\ \text{CONT : } \left[\begin{array}{l} \text{IND : } |3| \text{ref} \\ \text{RESTR|RELN : } pron \end{array} \right] \end{array} \right] \\ \text{DEPS : } |1| \oplus |2| \end{array} \right]$$

În M2 lista de complemente este goală ([COMPS: *el*]) deoarece argumentul obiect direct (indexat /2/) este realizat ca un afix, iar un afix nu poate avea proiecție sintagmatică. În schimb, afixul apare atât în lista de argumente (ARG-ST), cât și în cea de dependenți (DEPS). Acest lucru este corect: un *synsem* noncanonic poate fi un dependent sau (ca în cazul de față) atât un dependent, cât și un argument.

(iii’) În sfârșit, reprezentarea lexicală pentru structuri cu ACD arată astfel (M3):

¹⁷ CONT înseamnă “conținut”, IND înseamnă “index” și se referă la informația de persoană, număr și gen a numelui. Simbolul *ref* arată că un nume nu este expletiv. RESTR și RELN înseamnă “restricție” și, respectiv, “relație”. În sfârșit, *pron* înseamnă “pronominal”.

HEAD : <i>trans - cl - verb</i>								
VAL :	<table> <tr> <td>SUBJ : 1 </td> <td></td> </tr> <tr> <td>COMPS : 2 _{ss}</td> <td> <table> <tr> <td>HEAD : <i>noun</i> [MOD : 3 <i>aff - ss</i>]</td> </tr> <tr> <td>CONT RESTR RELN : <i>non - anaph</i></td> </tr> <tr> <td>MKING : <i>pe</i></td> </tr> </table> </td> </tr> </table>	SUBJ : 1		COMPS : 2 _{ss}	<table> <tr> <td>HEAD : <i>noun</i> [MOD : 3 <i>aff - ss</i>]</td> </tr> <tr> <td>CONT RESTR RELN : <i>non - anaph</i></td> </tr> <tr> <td>MKING : <i>pe</i></td> </tr> </table>	HEAD : <i>noun</i> [MOD : 3 <i>aff - ss</i>]	CONT RESTR RELN : <i>non - anaph</i>	MKING : <i>pe</i>
	SUBJ : 1							
COMPS : 2 _{ss}	<table> <tr> <td>HEAD : <i>noun</i> [MOD : 3 <i>aff - ss</i>]</td> </tr> <tr> <td>CONT RESTR RELN : <i>non - anaph</i></td> </tr> <tr> <td>MKING : <i>pe</i></td> </tr> </table>	HEAD : <i>noun</i> [MOD : 3 <i>aff - ss</i>]	CONT RESTR RELN : <i>non - anaph</i>	MKING : <i>pe</i>				
HEAD : <i>noun</i> [MOD : 3 <i>aff - ss</i>]								
CONT RESTR RELN : <i>non - anaph</i>								
MKING : <i>pe</i>								
ARG - ST : 1 ⊕ 2								
DEPS : 1 ⊕ 2 ⊕ 3								

Comentarii

- De vreme ce argumentele sunt aici *synsem*-uri noncanonice, ceea ce este un argument este și o valență.
- Complementele nu sunt anafore (*non-anaph*), ceea ce înseamnă că ele trebuie să fie pronume sau expresii referențiale.
- Se poate observa că lista de dependenți este mai bogată. Ea cuprinde un nou dependent, cliticul însuși. Acesta nu apare în lista de valențe (deoarece are un *synsem* noncanonic) și nici în cea a argumentelor (pentru că nu participă la fenomenele de legare). Cu toate acestea el este un *dependent* al verbului, de vreme ce acesta din urmă îl selectează și îi atribuie cazul acuzativ.
- În conformitate cu M3, valența indexată /2/ modifică *synsem*-ul afixului pronominal, însă nici o sintagmă nu este proiectată, ca o consecință a acestei relații de modificare, deoarece pronumele modificat nu are rol sintagmatic. Singura sintagmă proiectată ca urmare a realizării complementului este de tipul centru-complement.

6. Remarci finale

1. În română tranzitivitatea este un fenomen supus unor constrângeri riguroase și determinat de numeroase distincții:

- Distincția între verbe care impun obiectului direct un conținut de tipul *nonanimat* (sau un subtip al acestuia) și verbe care nu impun această restricție.
- Distincția între realizarea lexicală/sintagmatică a obiectului direct și cea clitică.
- Distincția între realizarea argumentală și cea nonargumentală a cliticului.
- Distincția între obiectele directe marcate și cele nemarcate.
- Distincția între GN modificatoare și cele nonmodificatoare.

Constrângerile (i)-(v) sunt aranjate în ordinea crescândă a gradului lor de specificitate. Cea dintâi pare universală și dă seama de faptul că verbele din prima categorie pot lua ca obiect direct pronominal doar pronume de persoana a treia:

(26) (a) Ion a îndeplinit **misiunea/ordinul/*copilul**. (b) Ion a îndeplinit-o./ Ion
l-a îndeplinit.

(c) * Ion m-/te-a îndeplinit.

Pronominalizarea obiectului direct, indiferent de persoană, este posibilă doar cu verbe care nu impun această restricție (vezi (27) (a)-(b)) și cu verbe care cer ca obiectul direct să fie marcat *animat* sau *uman* (vezi (28) (a)-(b)):

(27) (a) Ion așteaptă **un autobuz/un coleg**. (b) Ion mă/il așteaptă.

(28) (a) Poliția a arestat **demonstrații/*autobuzul**. (b) Poliția m/te/l/-a arestat.

A doua distincție – cea dintre realizarea lexicală și clitică a argumentului - explică faptul că româna, ca și franceza și italiana, are clitice argumentale.

A treia constrângere arată că, în plus, cliticele românești pot fi și dependenți nonargumentali.

A patra constrângere plasează româna în aceeași “familie” cu spaniola, în care unele obiecte directe sunt marcate (în spaniolă, de „prepoziția” *a*).

Ultima constrângere explică ce este un GN anticipat de un clitic: el este un GN care modifică acel clitic.

În cadrul teoriei HPSG, putem codifica toate aceste distincții mulțumită unui sistem ierarhic de constrângeri lexicale cu moștenire multiplă.

2. Conform analizei de mai sus, cliticul joacă un dublu rol: el poate fi un argument noncanonic al verbului, sau (când este modificat de un GN) devine, în plus, sursa de caz a GN anticipat. Faptul că își păstrează calitățile referențiale infirmă ipoteza (exprimată în Monachesi (2000)) că anticiparea obiectului direct în limba română este o formă de marcare a acordului dintre verb și obiectul său direct.

3. Faptul că cliticul-argument apare în lista de dependenți nu înseamnă că el poate fi supus extracției. Într-adevăr, într-o teorie caracterizată de un lexicalism puternic, așa cum este HPSG, afixe nu participă la procese și relații sintactice.

4. Considerăm că explicația dată în această lucrare fenomenului de ACD în română poate fi folosită și pentru alte limbi și dialecte care ACD (de exemplu

bulgara sau spaniola argentiniană vorbită în Rio de la Plata). Ne referim aici la relația de modificare dintre GN anticipat și clitic. Modificarea este o relație suficient de generală pentru a da seama de toate cazurile de anticipare clitică, indiferent de motivele care determină realizarea acestui tip de structură (bineînțeles, motive diferite de la o limbă la alta).

Bibliografie

- Barbu, A.-M** *Complexul verbal*, manuscris, 1998 (sub tipar, Studii și Cercetări Lingvistice)
- Borer, H.** *Parametric Syntax*, Foris, Dordrecht, 1984
- Borer, H.** (ed.), *Syntax and Semantics, vol.19, The Syntax of Pronominal Clitics*, Academic Pres, Inc., Harcourt Brace Jovanovich, Publishers, Orlando, 1986
- Black, J. R. and V. Montapanyane (eds.)** *Pronouns, Clitics and Movement*, John Benjamin, Publishing Company, Amsterdam / Philadelphia 1998
- Bouma, G. R. Malouf and I. A. Sag** *Satisfying Constraints on Extraction and Adjunction*, manuscris, 1999
- Balari, S. and L. Dini (eds.)** *Romance in HPSG*, CSLI, Stanford, 1997
- Cornilescu, A.** *A Note on Dative Clitics and Dative Case in Romanian*, Revue Roumaine de Linguistique, tome XXXII, No. 3, 1987, p. 213-225
- Dobrovie-Sorin, C.** *The Syntax of Romanian*, Mouton De Gruyter, Berlin, New York, 1994
- Farkas, D.** *Direct and Indirect Object Reduplication in Romanian*, Proceedings of the 14th Regional Meeting of the CLS, Chicago, 1978, p. 88-97
- Gerlach, B. and J. Grijzenhout (eds.)** *Clitics in Phonology, Morphology and Syntax*, John Benjamins Publishing Company, Amsterdam / Philadelphia, 2000
- Gierling, D.** *Clitic Doubling, Specificity and Focus in Romanian*, in Black and Montapanyane (1998), p.63-85
- Gutiérrez-Rexach, J.** *The Formal Semantics of Clitic Doubling*, manuscris, 1999
- Guțu-Romalo, V.** *Morfologie structurală a limbii române*, Editura didactică și pedagogică, București, 1968
- Ionescu, E.** *Accusative Weak Pronouns in Romanian*, in Cahiers de Linguistique Théorique et Appliquée, tomes XXXII-XXXIII, 1995-1996, p. 40-52
- Ionescu, E.** *On the Status of PE in the Direct Object Construction in Romanian*, manuscris, 2001 (sub tipar, Proceedings of the Romanian Academy)
- Jaeggli, O.** *Three Issues in the Theory of Clitics: Case, Doubled NPs, and Extraction*, in Borer (1986), p.15-42
- Miller, Ph. and I. A. Sag** *French Clitic Movement without Clitic or Movement*, Natural Language and Linguistic Theory, 15 (3), 1997, p. 573-639
- Monachesi, P.** *Decomposing Italian Clitics*, in Balari and Dini (1997) p. 301-354
- Monachesi, P.** *Clitic Placement in the Romanian Verbal Complex*, in Gerlach and Grijzenhout (2000), p. 255-294.
- Sanfilippo, A.** *Thematically Bound Adjuncts*, in Balari and Dini (1997) p. 355-391
- Pollard, C. and I. A. Sag** *Head-driven Phrase Structure Grammar*, The University of Chicago Press, Chicago, 1994

Buletinul RORIC-LING

Iunile 1 - 6

Au fost puse **141** de intrebari de catre utilizatori provenind in special din unitatile de invatamant romanesti, dar nu numai. Firmele de software atat din Romania, cat si din strainatate sunt, de asemenea, reprezentate. Unele dintre intrebari au fost puse de mai multe ori, asa cum va fi specificat in buletin. Exista patru mari categorii de intrebari, corespunzand tematicii partii romanesti a proiectului BALRIC-LING, dupa cum urmeaza:

- intrebari cu caracter general;
- intrebari referitoare la gramaticile de dependenta si la instrumentul de adnotare DGA;
- intrebari referitoare la relatiile de dependenta care au fost stabilite ca fiind tipice pentru limba romana, cat si referitoare la procesul de adnotare cu DGA a textelor romanesti;
- intrebari referitoare la formalismul HPSG.

In interiorul buletinului, intrebarile au fost grupate in conformitate cu tematica la care se refera (si nu in ordinea subscrierii, adica in ordine cronologica). Toate informatiile referitoare la numele si datele personale ale utilizatorilor exista in fisierele RORIC-LING, dar au fost sterse din buletin, pentru a facilita citirea si folosirea acestui material, precum si cautarea in cadrul lui dupa tematica.

Cateva statistici:

Intrebari: 141

Tara: **Romania Alta***
Intrebari: 107 34

Limba Materna: **Romana Alta**
Intrebari: 121 20

Domeniu de Activitate: **Educatie Cercetare Industrie Software Altul**
Intrebari: 89 18 12 22

• AUSTRALIA, AUSTRIA, CANADA, FRANCE, GERMANY, GREECE, ITALY, NETHERLANDS, TURKEY, UNITED KINGDOM, UNITED STATES

Intrebări cu caracter general

Care este scopul exact al acestei lucrări?

Din felul în care ati formulat întrebarea nu rezulta clar dacă va referiti la scopul unui anumit material publicat de Centrul nostru de Informare pe web sau la scopul întregului proiect. De aceea, ne permitem să vă răspundem într-un cadru mai larg. Dacă veți simți nevoia unor amănunte sau a unor clarificări ulterioare, vă rugăm să ne contactați din nou.

Obiectivul principal al proiectului BALRIC-LING este acela de a mari gradul de informare asupra potențialului celor mai avansate tehnologii privitoare la limbajul natural, în special în zona Balcanilor, unde domeniile procesării limbajului natural și al lingvisticii computaționale sunt mai puțin cunoscute. Întrucât HLT (Human Language Technologies) reprezintă un domeniu extrem de vast, BALRIC-LING se va concentra asupra a patru teme principale: resurse lingvistice organizate în jurul cuvântului, corpusuri și tagging, instrumente relevante pentru tratarea și realizarea acestora și posibile utilizări ale primelor două.

Pentru a ridica gradul de informare asupra acestor probleme în special în Bulgaria și România, în cadrul proiectului BALRIC-LING au fost înființate două Centre Regionale de Informare în aceste țări. Centrul de Informare românesc se numește RORIC-LING și el se va concentra asupra subtemelor menționate în pagina web a centrului, accesibilă la adresa de la care ati subscrib.

O problemă de larg interes actualmente este aceea a creării de resurse în general, a corpusurilor în mod special, fără de care realizarea unor aplicații avansate de tip HLT este imposibilă. În particular, în cadrul primei subteme, RORIC vă oferă un instrument de adnotare pentru crearea unui corpus, instrument care lucrează în cadrul formal al gramaticilor de dependență și a cărui valoare rezidă, în primul rând, în faptul că este independent de limbă. Sunt oferite numeroase exemple de folosire a lui în cazul limbii române (texte adnotate), care interesează în mod special pe utilizatorii din țara noastră.

Ceilalți parteneri ai proiectului se vor referi, în această perioadă, la o problemă diferită, care se încadrează în tematica menționată anterior. Pentru a afla și alte detalii referitoare la proiectul BALRIC-LING și a putea intra în paginile web ale celorlalți parteneri, puteți consulta și pagina de bază a proiectului, accesibilă la adresa <http://www.lml.bas.bg> de unde SE ALEGE BALRIC-LING. Vă mulțumim pentru subscriere și pentru interesul manifestat.

Hallo RORIC-LING-team! Un proiect interesant! Sunt informaticiana si intrebarile mele sunt: in ce masura am nevoie de alte cunostinte specifice (spre exemplu gramatica specifica a diferitelor limbi straine, etc. si cat de profund) pentru a ma putea aprofunda in acest proiect?

Consideram ca este suficienta intelegerea conceptelor si a teoriei prezentate de noi in materialul de pe web. Daca veti dori vreodata sa aplicati aceste teorii lingvistice referitor la o limba data, in cadrul unor aplicatii de procesare a limbajului natural (spre exemplu parsing), asa cum am facut-o noi pentru limba romana, va fi oricum nevoie sa apelati la consultanta lingvistica, care va va pune la dispozitie aplicarea teoriei referitor la limba respectiva. Pentru a putea afla unele informatii despre tipurile de aplicatii care exista, din punct de vedere informatic, va recomandam si consultarea buletinului virtual pe care il vom publica pe web la sfarsitul lunii februarie. Va multumim pentru interesul manifestat in legatura cu proiectul nostru.

Felicitari pentru proiectul indraznet in care sunteti implicati! Speram sa va descurcati cu succes. Acest proiect presupune si o procedura de procesare de text si, in caz afirmativ, care este abordarea pe care o veti alege: cea clasica, bazata pe calculul propozitional (Chomsky) sau tratarea ca pe o problema de clasificare de texte? Daca va fi utilizata alternativa ultima, care sunt strategiile de extragere a caracteristicilor si de invatare pe care intentionati sa le folositi?

Va multumim pentru subscriere si pentru interesul aratat fata de proiectul nostru. Proiectul nu presupune si o procedura de procesare de text, cel putin nu in acest stadiu de inceput. BALRIC-LING este, in principal, un proiect de ridicare a nivelului de cunoastere, avand ca principal obiectiv marirea gradului de informare privitor la HLT ("Humal Language Technologies") in special in zona Balcanilor. Prima parte a proiectului se concentreaza asupra resurselor lingvistice centrate in jurul cuvintului, corpusuri si tagging, precum si asupra instrumentelor relevante corespunzatoare. In cazul in care proiectul va fi prelungit, temele viitoare ar putea include o procedura de procesare a textelor, care, cel mai probabil, nu va presupune o abordare clasica "de tip Chomsky".

Care este conexiunea intre materialul existent si celelalte doua care urmeaza a fi expuse?

Conexiunea dintre aceasta parte a proiectului si ultima va deveni evidenta atunci cand ne vom ocupa de stabilirea unei specificatii teoretice pentru un model morfologic al limbii romane. Cea de-a doua tema propusa de RORIC si referitoare la WordNet reprezinta un subiect complet distinct, scopul seminarului virtual organizat de RORIC fiind acela de a supune dezbaterii o tematica diferita, dar care

se refera la cateva aspecte esentiale ale tehnologiei limbajului: resurse lingvistice si adnotari centrate in jurul cuvantului; corpusuri si tagging; instrumente relevante pentru tratarea si realizarea acestora.

Cat timp va dura acest proiect? Acest proiect de tip HLT se adreseaza numai companiilor de tip IT si persoanelor interesate in IT sau este adresat si publicului larg? Sunt interesat de obtinerea mai multor informatii legate de proiectul dvs. de tip HLT.

Proiectul BALRIC-LING a inceput la 1.09.2001 si va dura 18 luni (in afara cazului in care va fi prelungit). Proiectul este finantat de Comisia Europeana.

Obiectivul principal al proiectului BALRIC-LING este acela de a mari gradul de informare asupra potentialului celor mai avansate tehnologii privitoare la limbajul natural, precum si a posibilelor aplicatii de natura stiintifica si industrială a resurselor lingvistice corespunzatoare. Ridicarea gradului de cunoastere privitor la aceste aspecte este necesara in special in zona Balcanilor, unde domeniile procesarii limbajului natural si al lingvisticii computationale sunt mai putin cunoscute. Intrucat HLT reprezinta un domeniu extrem de vast, BALRIC-LING se va concentra asupra a patru teme principale: resurse lingvistice si adnotare organizate in jurul cuvantului, corpusuri si tagging, instrumente relevante pentru tratarea si realizarea acestora si posibile utilizari avansate de tip HLT ale primelor doua.

Pentru a ridica gradul de informare asupra acestor probleme in special in Bulgaria si in Romania, in cadrul proiectului BALRIC-LING au fost infiintate doua Centre Regionale de Informare in aceste tari. Centrul de informare romanesc se numeste RORIC-LING si el se va concentra asupra subtemelor mentionate in pagina web a centrului, accesibila la adresa de la care ati subscriis.

Incepand de astazi veti gasi mai multe detalii privitoare la intregul proiect BALRIC-LING si in pagina de baza a Centrului de Informare RORIC-LING.

Ceilalti parteneri ai proiectului se vor referi la o problematica diferita, care se incadreaza in tematica generala mentionata anterior. Nu toti partenerii si-au incarcat inca paginile web, dar o vor face in curand. Pentru a afla si alte detalii referitoare la proiectul BALRIC-LING si a putea intra in paginile web ale celorlalti parteneri, puteti consulta si pagina de baza a proiectului, accesibila la adresa <http://www.lml.bas.bg> de unde se alege BALRIC-LING.

Proiectul nu se refera numai la zona Balcanilor, ci se adreseaza persoanelor interesate de HLT de pretutindeni. De asemenea, el nu se deruleaza numai in beneficiul companiilor si al persoanelor interesate de IT. Proiectul isi propune sa

mareasca gradul general de informare asupra acestor domenii. Vom fi incantati sa raspundem intrebărilor provenind de la toti cei care sunt - sau devin - interesati.

Va multumim pentru subscriere si pentru interesul manifestat.

Intrebări referitoare la gramaticile de dependenta si DGA

Care sunt diferentele esentiale dintre gramaticile de dependenta si cele generative?

Exista câteva diferente majore între D-limbaj (limbajul gramaticilor de dependenta) si PS-limbaj (limbajul gramaticilor PS). Precizam ca, în cele ce urmeaza, arborele de derivare rezultat în urma efectuării analizei sintactice care utilizeaza o gramatica PS va fi denumit PS-arbore, în timp ce arborele corespunzător rezultat în urma utilizării în analiza sintactica a unei gramatici de dependenta va fi numit D-arbore.

O prima diferenta semnificativa între D-limbaj si PS-limbaj consta în aceea ca un PS-arbore corespunzător unei expresii aparținând limbajului natural arata care elemente ale acesteia (cuvinte sau chiar grupuri sintactice) se pot combina cu alte elemente pentru a forma niste unitati de ordin mai mare. Un PS-arbore dezvaluie structura unei propozitii în termeni de grupari ale elementelor sale: blocuri maximale care constau din blocuri mai mici, care, la rândul lor, constau din blocuri si mai mici etc. PS-structura se exprima în termeni de constituenți, operatia logica aflata la baza acestei abordari fiind aceea a incluziunii de multimi, cu ajutorul careia se exprima apartenenta la un grup sintactic, la o categorie etc.. Aceasta abordare favorizeaza punctul de vedere analitic. Un D arbore, pe de alta parte, arata ce elemente se afla în relatie cu alte elemente si în ce mod. D-structura propozitiei reflecta relatiile existente între unitati sintactice indivizibile, lucrând direct cu forme lexicale. În aceasta abordare, operatia logica de baza este aceea a stabilirii de relatii binare. Propozitia nu mai este alcatuita din grupuri sintactice, categorii, ci din cuvinte legate între ele prin relatii de dependenta. Aceasta abordare favorizeaza, prin urmare, punctul de vedere sintetic.

O alta diferenta între PS-limbaj si D-limbaj este data de faptul ca, în cadrul unui PS-arbore, apartenenta la o anumita categorie este specificata ca parte a reprezentării sintactice. Simboluri ca NP, VP, N etc. intervin în PS arbori ca etichete ale unor vârfuri. Cu alte cuvinte, unele caracteristici sintactice date de operatii precum categorizarea si subcategorizarea sunt folosite ca instrument principal în exprimarea rolului sintactic. În cadrul unui D-arbore, pe de alta parte, simbolurile reprezentând apartenenta la o categorie, precum si alte proprietati sintactice nu sunt admise ca elemente imediate ale structurii sintactice. (Astfel de informatii sunt incluse în dictionar, lexicon etc.).

O a treia diferență esențială constă în faptul că, într-un PS-arbore, majoritatea nodurilor corespund unor simboluri neterminale. Ele reprezintă grupuri sintactice și nu corespund formelor lexicale efective care intervin în propoziția analizată. Prin contrast, un D-arbore conține numai noduri terminale, nefiind necesară nici o reprezentare abstractă a grupurilor sintactice.

PS-limbajul este, în esență, un limbaj linear, în timp ce D-limbajul este unul bidimensional, aceasta generând o altă deosebire fundamentală între cele două tipuri de reprezentări sintactice discutate aici. Astfel, în cadrul unui PS-arbore, vârfurile arborelui trebuie să fie ordonate linear, ordinea nefiind neapărat cea a formelor lexicale care intervin în propoziție. În cadrul unui D-arbore, pe de altă parte, vârfurile nu se află într-o astfel de ordine. Ordinea liniară a formelor lexicale din interiorul propoziției este un mijloc folosit de limbile naturale pentru a codifica relații sintactice și, prin urmare, ordinea liniară nu trebuie să intervină în structurile sintactice.

În fine, o ultimă deosebire importantă între cele două tipuri de reprezentări constă în aceea că, în timp ce un PS-arbore nu specifică tipul legăturii sintactice existente între doi constituenți, un D-arbore pune în mod special accentul pe specificarea în detaliu a tipului legăturii dintre oricare două elemente aflate în relație de dependență.

Cum diferă DG (Dependency Grammar) de PSG (Phrase-Structure Grammar)?

Așa cum arată Richard Hudson, gramaticile, precum și teoriile gramaticale, pot fi clasificate în funcție de unitatea de bază a structurii propoziției. Clasificarea se face după cum la baza structurii propoziției se află

- grupul sintactic ("the phrase" - PSG);
- dependența dintre două cuvinte (DG).

Fiecare abordare o implică pe cealaltă:

- PSG presupune existența unor dependente între cuvinte (dar numai în sensul în care un cuvânt este desemnat ca fiind capul grupului sintactic, adică centrul în jurul căruia se organizează acest grup);
- DG face trimitere la grupuri sintactice (un cuvânt împreună cu cuvintele care depind de el și cu grupurile sintactice ale acestora formează un grup sintactic).

Este DG ("Dependency Grammar") doar o varianta notationala a lui PSG ("Phrase-Structure Grammar")?

Un mare numar de logicieni, printre care si Bar-Hillel, au demonstrat ca DG (inclusiv gramatica categoriala) este SLAB echivalenta cu o gramatica PSG independenta de context (Gaifman, "Dependency systems and phrase-structure systems"). Dar ea NU reprezinta o varianta notationala relativ la PSG, intrucat nu este PUTERNIC echivalenta i.e., asa cum arata Richard Hudson, nu permite aceleasi analize:

- grupurile sintactice sunt implicite, nu explicite, asa incat
 - grupurile sintactice nu pot fi clasificate separat de cuvintele cap corespunzatoare;
- relatiile sunt explicite, nu implicite, deci
 - relatiile pot fi clasificate si etichetate;
- toate grupurile sintactice trebuie sa fie endocentrice, deci
 - constructii in aparenta exocentrice cum ar fi gerunziile reprezinta o dificultate fundamentala;
- nu sunt permise nodurile neterminale, prin urmare DG nu permite
 - ramificarea unara (de ex. un grup sintactic de tip NP care consta numai dintr-un substantiv).

Care au aparut mai intai, gramaticile de dependenta sau cele generative? Faceti un scurt istoric.

Etapele principale in evolutia celor doua tipuri de gramatici ar fi urmatoarele:

1. Panini (acum 2600 de ani ;India) a recunoscut si clasificat dependentele semantice, sintactice si morfologice;
2. lingvistii arabi (acum 1200 de ani; Irak) au recunoscut structura sintactica de dependenta;
3. lingvistii latini (acum 800 de ani) au recunoscut "determinarea" si structurile de dependenta;
4. scolile lingvistice referitoare la limba engleza din Europa si S.U.A. au predat analiza propozitiilor in termeni de dependenta, iar diagramele concepute pentru propozitii care au devenit extrem de cunoscute la sfarsitul

sec. al XIX-lea (și care foloseau un sistem inventat în S.U.A.) erau de tip DG.

5. Lucien Tesnière (Franta, 1930) a dezvoltat o teorie formală și relativ sofisticată a gramaticilor de dependență pentru folosire în școli. Această abordare de tip "bottom-up" este încă folosită atât în Europa, cât și în S.U.A.
6. În 1933 Leonard Bloomfield din S.U.A. dezvoltă o abordare de tip "top-down": analiza constituentilor imediați, care se va transforma în "analiza PS" (de la "phrase-structure analysis").

Popularitatea dependentelor, ca mijloc formal de reprezentare a structurii sintactice a propozițiilor, a fost mereu în creștere și a culminat cu opera lui Lucien Tesnière din 1959. Cu toate acestea, în America de Nord, la începutul anilor '30, "sintaxa de dependență" a fost eclipsată de ceea ce s-a numit, la acea vreme, "sintaxa constituentilor imediați" (sau "analiza de tip IC" - de la "immediate constituency"). Aceasta s-a transformat mai târziu în "analiza PS", care determină PS-structura unei propoziții. Formulată în mod riguros de Leonard Bloomfield (Bloomfield 1933), dar și de către Wells în 1947 și Percival în 1976, reprezentarea de tip PS în sintaxă a fost promovată cu multă energie de școala structuralistă în anii '30, '40 și '50. Ea a devenit unica reprezentare sintactică luată în considerare de către Noam Chomsky și școala generativ transformatională pe care acesta a fondat-o la sfârșitul anilor '50.

Care dintre cele 2 clase de gramatici (de dependență și respectiv generative) surprind cel mai bine fenomenele din limbajul natural?

Răspunsul depinde de ce se înțelege prin gramatici generative. Clasa gramaticilor generative este foarte largă, în interiorul acestei clase propunându-se diverse formalisme care să surprindă fenomenele limbajului natural. De asemenea și gramaticile de dependență au fost formalizate în diverse moduri.

Vom încerca să răspundem la această întrebare din trei puncte de vedere, și anume:

1. Din punct de vedere formal. Acest punct de vedere privește capacitatea generativă a unei clase de gramatici. Pentru a fi considerată adecvată o clasă de gramatici trebuie să fie suficient de restrictivă astfel încât să nu permită generarea (descrierea) oricărui tip de limbaj, dar și suficient de puternică pentru a permite descrierea fenomenelor întâlnite în limbajul natural. Din acest punct de vedere considerăm că cele două clase de gramatici sunt echivalente.

După ce s-a acceptat faptul că fenomenele din limbajul natural depășesc capacitatea descriptivă a gramaticilor independente de context, în ultimul timp s-a conturat clasa limbajelor

semidependente de context (mildly context-sensitive languages), care este in general acceptata ca fiind suficienta pentru descrierea limbajului natural si care este generata de mai multe formalisme gramaticale (propuise independent si din ratiuni diferite):

K. Vijay-Shanker, D.J. Weir, The Equivalence of Four Extensions of Context-Free Grammar. Math. Systems Theory, 27, 1994.

Pentru gramaticile de dependenta exista formalizari care le fac echivalente cu gramaticile independente de context:

H. Gaifman, Dependency systems and phrase-structure systems. Information & Control, 8, 1965.

dar si formalizari care le permit sa descrie limbajele semidependente de context:

H. Maruyama, Constraint dependency grammar and its weak generative capacity. Computer Software, 1990.

2. Din punct de vedere lingvistic. Acest punct de vedere priveste usurinta cu care un lingvist poate descrie fenomene lingvistice specifice unei limbi in cadrul unui anumit formalism. Din acest punct de vedere credem ca raspunsul depinde de limba care se are in vedere si de traditia lingvistica specifica limbii respective. Pentru limba romana consideram mai adecvat formalismul gramaticilor de dependenta deoarece este mai apropiat de modul traditional de analiza sintactica al limbii romane, acest lucru permitand inglobarea mai usoara a cunostintelor puse la dispozitie de lingvistica romaneasca.
3. Din punctul de vedere al modelarii stochastice a limbajului. Pentru o discutie privind avantajele gramaticilor de dependenta in modelarea stochastica a limbajului natural a se vedea sectiunea 12.1.7 din

C. D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing. The MIT Press, 1999.

Aici nu vom mentiona decat faptul ca sistemul de analiza sintactica (parsing) stochastica cu cele mai bune performante pana la ora actuala este bazat pe gramatici de dependenta:

M. J. Collins, Three generative, lexicalised models for statistical parsing. ACL 35, 1997.

Care sunt principalele teorii bazate pe notiunea de dependenta si avand la baza gramaticile de dependenta?

Principalele teorii bazate pe notiunea de dependenta sunt urmatoarele:

- Case Grammar (Anderson)
- Daughter-Dependency Theory (Hudson)
- Dependency Unification Grammar (Hellwig)
- Functional-Generative Description (Sgall)
- Lexicase (Starosta)
- Meaning-Text Model (Melcuk)
- Metataxis (Schubert)
- Unification Dependency Grammar (Maxwell)
- Constraint Dependency Grammar (Maruyama)

Este "Link Grammar" o gramatica de tip "Dependency Grammar"? (pusa de trei ori)

Link Grammar (introdusa de Daniel D. Sleator si Davy Temperley) este "de tip dependency", dar mult mai lexicalizata. Un astfel de formalism gramatical presupune ca o succesiune de cuvinte apartine limbajului generat de o Link Grammar daca exista o modalitate de a crea legaturi intre cuvinte astfel incat: (1) cerintele locale ale fiecarui cuvant sunt satisfacute, (2) legaturile (arcele) nu se intersecteaza si (3) cuvintele formeaza un graf conex. Formalismul este lexical si nu utilizeaza in mod explicit constitienti si categorii.

Link Grammars se aseamana cu gramaticile de dependenta si cu gramaticile categoriale. Exista si multe diferente semnificative, cel mai important aspect fiind acela ca Link Grammar este o gramatica mult mai lexicalizata.

Ce este Word Grammar? Este acest tip de gramatica inrudit cu Dependency Grammar?

Word Grammar este o teorie gramaticala dezvoltata de Richard Hudson inca de la inceputul anilor '80. Teoria se bazeaza in mod strans pe DG si prezinta, dupa toate probabilitatile, cea mai buna combinatie posibila a altor caracteristici. Cele mai importante caracteristici ale WG, asa cum sunt ele consemnate de autorul acestei teorii gramaticale, sunt urmatoarele:

- este monostratala - corespunzator unei propozitii exista o unica structura sintactica, imperecheata cu o structura semantica si cu una fonologica;
- este imbogatita - permite dependente multiple (un cuvant poate avea mai multe "cuvinte parinte");

- generalizeaza, prin intermediul mostenirii implicite bazate pe relatia de tip "isa";
- permite relatii etichetate (intr-o ierarhie de tip "isa");
- este nemodulara si cognitiva: limba este o zona a retelei generale de cunostinte.

Ce alte tipuri de gramatici sunt folosite in NLP?

Algoritmii de analiza sintactica clasici de tip "top-down" si respectiv "bottom-up" sunt bazati pe gramatici generative, care privesc structura propozitiei ca fiind alcatuita din constitienti. In acest caz, structura unei propozitii, data de constituentii sai, reprezinta conceptul central al sintaxei. Spre deosebire de gramaticile generative, gramaticile de dependenta nu se bazeaza pe notiunea de constituent, ci pe relatii directe existente intre cuvinte. Structura de dependenta poate fi privita, printre altele, ca opunandu-se structurii alcatuite din constitienti. Ideea centrala pe care se bazeaza notiunea de dependenta este aceea ca fiecare cuvant este privit ca depinzand de cuvantul care il leaga de restul propozitiei, practic explicand de ce este utilizat. Spre deosebire de gramaticile generative, cele de dependenta pot descrie cu mai mult succes fenomene lingvistice cum ar fi existenta constituentilor discontinui sau variatia ordinii cuvintelor in cadrul propozitiei.

O alta clasa de gramatici care genereaza limbaje ce nu au o legatura directa cu ierarhia Chomsky (neputand fi comparate cu familiile de baza ale acestei ierarhii) este aceea a gramaticilor contextuale. Gramaticile contextuale au fost introduse de Solomon Marcus in 1969. Acesta le introduce ca pe niste "gramatici intrinseci", fara simboluri auxiliare, bazate numai pe operatia lingvistica fundamentala de inserare a cuvintelor in structuri date, in conformitate cu anumite dependente contextuale. Gramaticile contextuale includ contexte (sau perechi de cuvinte) asociate unor selectori (multimi de cuvinte). Un context poate fi alaturat oricarui element selector asociat. In acest fel, pornindu-se de la o multime finita de cuvinte (axiome), este generat limbajul. S-a aratat ca acest formalism modeleaza foarte bine limbajul natural. De abia in 1999 K. Harbusch reuseste sa prezinte un parser bazat pe gramatici contextuale. Rezultate recente extrem de incurajatoare i-au determinat pe cercetatori sa se concentreze asupra construirii unei gramatici contextuale a limbii engleze.

Alte tipuri de gramatici folosite in NLP sunt "Link Grammars" si "Tree Adjoining Grammars". Va rugam sa ne contactati din nou daca sunteti interesat de aflarea unor detalii referitor la o anumita clasa de gramatici.

Cunoasteti un alt exemplu de gramatica (in afara de gramatica de dependenta) care modeleaza foarte bine limbajul natural?

O alta clasa de gramatici care modeleaza foarte bine limbajul natural si care genereaza limbaje ce nu au o legatura directa cu ierarhia Chomsky (neputand fi comparate cu familiile de baza ale acestei ierarhii) este aceea a gramaticilor contextuale. Gramaticile contextuale au fost introduse de romanul Solomon Marcus in 1969. Acesta le introduce ca pe niste "gramatici intrinseci", fara simboluri auxiliare, bazate numai pe operatia lingvistica fundamentala de inserare a cuvintelor in structuri date, in conformitate cu anumite dependente contextuale. Gramaticile contextuale includ contexte (sau perechi de cuvinte) asociate unor selectori (multimi de cuvinte). Un context poate fi alaturat oricarui element selector asociat. In acest fel, pornindu-se de la o multime finita de cuvinte (axiome), este generat limbajul. S-a aratat ca acest formalism modeleaza foarte bine limbajul natural. De abia in 1999 K. Harbusch reuseste sa prezinte un parser bazat pe gramatici contextuale. Rezultate recente extrem de incurajatoare i-au determinat pe cercetatori sa se concentreze asupra construirii unei gramatici contextuale a limbii engleze. Pentru mai multe informatii asupra gramaticilor contextuale, vezi

S.Marcus, C.Martin-Vide, G.Paun. Contextual Grammars as Generative Models of Natural Languages. Computational Linguistics, 24(2), p. 245-274.

F.Hristea. Introducere in procesarea limbajului natural cu aplicatii in Prolog. Editura Universitatii din Bucuresti, 2000, p. 102-113.

Gramaticile de dependenta si gramaticile contextuale sunt unul si acelasi tip de gramatici?

NU. Gramaticile contextuale reprezinta o alta clasa de gramatici care modeleaza foarte bine limbajul natural si care genereaza limbaje ce nu au o legatura directa cu ierarhia Chomsky (neputand fi comparate cu familiile de baza ale acestei ierarhii). Gramaticile contextuale au fost introduse de romanul Solomon Marcus in 1969. Acesta le introduce ca pe niste "gramatici intrinseci", fara simboluri auxiliare, bazate numai pe operatia lingvistica fundamentala de inserare a cuvintelor in structuri date, in conformitate cu anumite dependente contextuale. Gramaticile contextuale includ contexte (sau perechi de cuvinte) asociate unor selectori (multimi de cuvinte). Un context poate fi alaturat oricarui element selector asociat. In acest fel, pornindu-se de la o multime finita de cuvinte (axiome), este generat limbajul. S-a aratat ca acest formalism modeleaza foarte bine limbajul natural. De abia in 1999 K. Harbusch reuseste sa prezinte un parser bazat pe gramatici contextuale. Rezultate recente extrem de incurajatoare i-au determinat pe cercetatori sa se concentreze asupra construirii unei gramatici contextuale a limbii engleze.

Ce algoritmi de analiza sintactica (parsing) exista pentru gramaticile de dependenta? Grupul dvs. a folosit vreunul pana in prezent?

In cadrul formal al gramaticilor de dependenta s-a efectuat analiza sintactica folosind "Constraint Dependency Grammar" (Maruyama, 1990). CDG face o separare clara intre posibilele descrieri structurale si conditiile de corectitudine pentru structurile lingvistice. CDG este slab dependenta de context. Pentru a citi despre algoritmi bazati pe CDG va recomandam consultarea lucrarii:

Menzel, Wolfgang si Schroder, Ingo, "Decision procedures for dependency parsing using graded constraints", in: Sylvain Kahane si Alain Polguere (editori), "Proc. Coling - ACL Workshop on Processing of Dependency-based Grammars", pag. 78-87, Montreal, Canada, 1998.

In ceea ce priveste grupul nostru, am efectuat analiza sintactica de dependenta intr-o abordare stocastica, in care nu este necesara specificarea unei gramatici de dependenta propriu-zise. Gramatica a fost in mod implicit inclusa in parametrii modelului stocastic, care, la rindul lor, au fost estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru, a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii din materialul publicat pe web.

Gasirea multimii T s-a facut utilizandu-se un algoritm propus de Ratnaparkhi in 1996. Acest algoritm este de natura stocastica si utilizeaza entropia maxima. Gasirea multimii P s-a facut, de asemenea, prin utilizarea unui algoritm stocastic, si anume a algoritmului lui Eisner, propus in acelasi an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezinta o implementare a metodei programarii dinamice cu scopul de a gasi cea mai probabila analiza in maniera "bottom-up" (de jos in sus). Dupa determinarea multimilor T si P, gasirea multimii D nu mai ridica nici un fel de probleme.

Ce algoritmi de analiza sintactica (parsing) exista pentru gramaticile de dependenta?

In cadrul formal al gramaticilor de dependenta s-a efectuat analiza sintactica folosind "Constraint Dependency Grammar" (Maruyama, 1990). CDG face o separare clara intre posibilele descrieri structurale si conditiile de corectitudine

pentru structurile lingvistice. CDG este slab dependenta de context. Pentru a citi despre algoritmi bazati pe CDG va recomandam consultarea lucrarii:

Menzel, Wolfgang si Schroder, Ingo, "Decision procedures for dependency parsing using graded constraints", in: Sylvain Kahane si Alain Polguere (editori), "Proc. Coling - ACL Workshop on Processing of Dependency-based Grammars", pag. 78-87, Montreal, Canada, 1998.

Grupul nostru a efectuat, la randul sau, analiza sintactica de dependenta. Abordarea noastra a fost una stocastica, in care nu este necesara specificarea unei gramatici de dependenta propriu-zise. Gramatica a fost in mod implicit inclusa in parametrii modelului stocastic, care, la rindul lor, au fost estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru, a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii din materialul publicat pe web.

Gasirea multimii T s-a facut utilizandu-se un algoritm propus de Ratnaparkhi in 1996. Acest algoritm este de natura stocastica si utilizeaza entropia maxima. Gasirea multimii P s-a facut, de asemenea, prin utilizarea unui algoritm stocastic, si anume a algoritmului lui Eisner, propus in acelasi an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezinta o implementare a metodei programarii dinamice cu scopul de a gasi cea mai probabila analiza in maniera "bottom-up" (de jos in sus). Dupa determinarea multimilor T si P, gasirea multimii D nu mai ridica nici un fel de probleme.

Cunoasteti un parser bazat pe gramatici de dependenta? Ati facut vreodata analiza sintactica bazata pe gramatici de dependenta?

Grupul nostru a efectuat analiza sintactica de dependenta intr-o abordare stocastica, in care nu este necesara specificarea unei gramatici de dependenta propriu-zise. Gramatica a fost in mod implicit inclusa in parametrii modelului stocastic, care, la rindul lor, au fost estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru, a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii din materialul publicat pe web. Etapele in derularea unui asemenea algoritm sunt: gasirea multimii T ("part of speech tagging"); gasirea multimii P (adica a relatiilor de dependenta); gasirea multimii D (adica a tipului dependentelor).

Gasirea multimii T s-a facut utilizandu-se un algoritm propus de Ratnaparkhi in 1996. Acest algoritm este de natura stocastica si utilizeaza entropia maxima. Gasirea multimii P s-a facut, de asemenea, prin utilizarea unui algoritm stocastic, si anume a algoritmului lui Eisner, propus in acelasi an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezinta o implementare a metodei programarii dinamice cu scopul de a gasi cea mai probabila analiza in maniera "bottom-up" (de jos in sus). Dupa determinarea multimilor T si P, gasirea multimii D nu mai ridica nici un fel de probleme.

Grupul dvs. a facut vreodata analiza sintactica (parsing) in contextul gramaticilor de dependenta si cum?

Grupul nostru a efectuat analiza sintactica de dependenta intr-o abordare stocastica, in care nu este necesara specificarea unei gramatici de dependenta propriu-zise. Gramatica a fost in mod implicit inclusa in parametrii modelului stocastic, care, la rindul lor, au fost estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru, a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii din materialul publicat pe web. Etapele in derularea unui asemenea algoritm sunt: gasirea multimii T ("part of speech tagging"); gasirea multimii P (adica a relatiilor de dependenta); gasirea multimii D (adica a tipului dependentelor).

Gasirea multimii T s-a facut utilizandu-se un algoritm propus de Ratnaparkhi in 1996. Acest algoritm este de natura stocastica si utilizeaza entropia maxima. Gasirea multimii P s-a facut, de asemenea, prin utilizarea unui algoritm stocastic, si anume a algoritmului lui Eisner, propus in acelasi an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezinta o implementare a metodei programarii dinamice cu scopul de a gasi cea mai probabila analiza in maniera "bottom-up" (de jos in sus). Dupa determinarea multimilor T si P, gasirea multimii D nu mai ridica nici un fel de probleme.

Grupul dvs. a adus ceva nou in teoria gramaticilor de dependenta sau in modul de folosire a acestora?

Grupul nostru a efectuat parsing stocastic in contextul gramaticilor de dependenta. Am efectuat analiza sintactica de dependenta intr-o abordare stocastica, in care nu este necesara specificarea unei gramatici de dependenta propriu-zise. Gramatica a fost in mod implicit inclusa in parametrii modelului stocastic, care, la rindul lor, au fost estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru, a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii din materialul publicat pe web. Etapele in derularea unui asemenea algoritm sunt: gasirea multimii T ("part of speech tagging"); gasirea multimii P (adica a relatiilor de dependenta); gasirea multimii D (adica a tipului dependentelor).

Gasirea multimii T s-a facut utilizandu-se un algoritm propus de Ratnaparkhi in 1996. Acest algoritm este de natura stocastica si utilizeaza entropia maxima. Gasirea multimii P s-a facut, de asemenea, prin utilizarea unui algoritm stocastic, si anume a algoritmului lui Eisner, propus in acelasi an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezinta o implementare a metodei programarii dinamice cu scopul de a gasi cea mai probabila analiza in maniera "bottom-up" (de jos in sus). Dupa determinarea multimilor T si P, gasirea multimii D nu mai ridica nici un fel de probleme.

Recomandati analiza sintactica de tip stocastic bazata pe gramatici de dependenta sau pe gramatici generative?

Recomandam analiza sintactica de tip stocastic bazata pe gramatici de dependenta intrucat ea a fost deja efectuata cu succes in cazul limbii romane. Astfel, grupul nostru a efectuat analiza sintactica de dependenta intr-o abordare stocastica, in care nu este necesara specificarea unei gramatici de dependenta propriu-zise. Gramatica a fost in mod implicit inclusa in parametrii modelului stocastic, care, la rindul lor, au fost estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru, a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii din materialul publicat pe web. Etapele in derularea unui asemenea algoritm sunt: gasirea multimii T ("part of speech tagging"); gasirea multimii P (adica a relatiilor de dependenta); gasirea multimii D (adica a tipului dependentelor).

Gasirea multimii T s-a facut utilizandu-se un algoritm propus de Ratnaparkhi in 1996. Acest algoritm este de natura stocastica si utilizeaza entropia maxima. Gasirea multimii P s-a facut, de asemenea, prin utilizarea unui algoritm stocastic, si anume a algoritmului lui Eisner, propus in acelasi an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezinta o implementare a metodei programarii dinamice cu scopul de a gasi cea mai probabila analiza in maniera "bottom-up" (de jos in sus). Dupa determinarea multimilor T si P, gasirea multimii D nu mai ridica nici un fel de probleme.

Considerati ca limba romana se preteaza mai bine la o abordare cu gramatici de dependenta decat la una cu gramatici de tip "phrase structure grammars"? (pusa de doua ori)

Da, intrucat aceasta abordare este mai apropiata de gramatica traditionala. Probabil ca acest lucru face ca lingvistii romani sa fie mult mai apropiati de aceasta abordare, pe care au si aplicat-o cu succes relativ la limba romana, efectuand analiza sintactica de dependenta, in cadrul proiectului DBR-MAT, finantat de Fundatia Volkswagen (1996-1998).

Ce ar presupune definirea unei gramatici de dependenta pentru o anumita limba? Exista o astfel de gramatica pentru limba romana?

In cadrul de lucru oferit de aceasta teorie lingvistica, specificarea unei gramatici de dependenta inseamna gasirea unei multimi de constrangeri care sa ajute la stabilirea faptului ca anumite structuri sintactice sunt corecte, iar altele nu. Spre exemplu, in virtutea unor asemenea constrangeri, se poate decide faptul ca anumite cuvinte ale unei propozitii pot avea rolul de cuvant cap, in timp ce altele nu pot detine acest rol. Cu alte cuvinte, specificarea unei gramatici de dependenta pentru o anumita limba inseamna stabilirea unor reguli care sa specifice ce relatii de dependenta sunt permise in limba respectiva. Pentru limba romana nu au fost stabilite aceste reguli, prin urmare nu exista o gramatica de dependenta, ci exista numai relatii de dependenta ca atare, ce pot fi folosite la diverse sarcini, cum ar fi efectuarea analizei sintactice de dependenta.

Exista vreo gramatica de dependenta pentru limba romana?

In cadrul de lucru oferit de aceasta teorie lingvistica, specificarea unei gramatici de dependenta inseamna gasirea unei multimi de constrangeri care sa ajute la stabilirea faptului ca anumite structuri sintactice sunt corecte, iar altele nu. Spre exemplu, in virtutea unor asemenea constrangeri se poate decide faptul ca anumite cuvinte ale unei propozitii pot avea rolul de cuvant cap, in timp ce altele nu pot detine acest rol. Cu alte cuvinte, specificarea unei gramatici de dependenta pentru o anumita limba inseamna stabilirea unor reguli care sa specifice ce relatii de dependenta sunt permise in limba respectiva. Pentru limba romana nu au fost stabilite aceste reguli, prin urmare nu exista o gramatica de dependenta, ci exista numai relatii de dependenta ca atare, ce pot fi folosite la diverse sarcini, cum ar fi efectuarea analizei sintactice de dependenta.

Dati un exemplu de utilizare a formalismului gramaticilor de dependenta in cazul limbii romane.

Formalismul gramaticilor de dependenta a fost utilizat, in cazul limbii romane, pentru realizarea analizei sintactice de tip stocastic (in cadrul proiectului DBR-MAT). In acest cadru formal putem spune ca a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii ca in materialul prezentat. Etapele in derularea unui asemenea algoritm sunt: gasirea multimii T ("part of speech tagging"), gasirea multimii P (adica a relatiilor de dependenta) si gasirea multimii D (adica a tipului dependentelor). Principala concluzie care s-a desprins, in cadrul proiectului DBR-MAT, dar independent de limba, a fost aceea ca formalismul gramaticilor de dependenta este extrem de adecvat si poate fi utilizat cu succes in efectuarea analizei sintactice de tip stocastic.

Exista corpusuri pentru limba romana cu texte analizate in formalismul gramaticilor de dependenta? (pusa de doua ori)

Crearea unui asemenea corpus a fost inceputa acum, in cadrul proiectului BALRIC LING. Textele existente deja pe web fac parte din acest corpus, iar numarul lor va creste ulterior.

Care sunt posibilele aplicatii software ale subiectelor prezentate (e.g. gramatici de dependenta)? Exista programe referitoare la limba engleza care ar putea fi preluate pentru limba romana de indata ce reguli/descrieri/adnotari corespunzatoare au fost definite in cazul limbii romane?

Un exemplu de aplicatie bazata pe gramatici de dependenta il constituie analiza sintactica, aplicatiile software fiind reprezentate de algoritmii de parsing corespunzatori.

In cadrul formal al gramaticilor de dependenta s-a efectuat analiza sintactica folosind "Constraint Dependency Grammar" (Maruyama, 1990). CDG face o separare clara intre posibilele descrieri structurale si conditiile de corectitudine pentru structurile lingvistice. CDG este slab dependenta de context. Pentru a citi despre algoritmi bazati pe CDG va recomandam consultarea lucrarii:

Menzel, Wolfgang si Schroder, Ingo, "Decision procedures for dependency parsing using graded constraints", in: Sylvain Kahane si Alain Polguere (editori), "Proc. Coling - ACL Workshop on

În ceea ce privește grupul nostru, am efectuat analiza sintactică de dependență într-o abordare stocastică, în care nu este necesară specificarea unei gramatici de dependență propriu-zise. Gramatica a fost în mod implicit inclusă în parametrii modelului stocastic, care, la rândul lor, au fost estimați pe baza datelor lingvistice (adică a unui corpus).

În acest cadru, a găsi un algoritm de analiză sintactică înseamnă a găsi un algoritm care are ca input o propoziție și ca output structura sintactică (S,D) a acelei propoziții, unde $S=(T,P)$ și D au aceleași semnificații din materialul publicat pe web.

Gasirea multimii T s-a făcut utilizându-se un algoritm propus de Ratnaparkhi în 1996. Acest algoritm este de natură stocastică și utilizează entropia maximă. Gasirea multimii P s-a făcut, de asemenea, prin utilizarea unui algoritm stocastic, și anume a algoritmului lui Eisner, propus în același an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezintă o implementare a metodei programării dinamice cu scopul de a găsi cea mai probabilă analiză în maniera "bottom-up" (de jos în sus). După determinarea multimilor T și P, gasirea multimii D nu mai ridică nici un fel de probleme.

Programele existente sunt independente de limbă și au fost testate de noi cu succes în cazul limbii române.

Care sunt avantajele folosirii DGA?

Principalele avantaje ale folosirii DGA deriva din faptul că programul reprezintă un instrument independent de limbă. El a fost, în egală măsură, proiectat pentru a fi independent de variantele de formalizare ale gramaticilor de dependență. Alte avantaje importante ale DGA deriva din caracteristicile sale, menționate în "manualul utilizatorului": ușurința în folosire, portabilitate, conformitate cu standardele actuale, flexibilitate.

În ce mod ar putea fi folosit un corpus obținut prin adnotare cu DGA?

Una dintre utilizările unui asemenea corpus o reprezintă efectuarea analizei sintactice (parsing). Acest lucru a și fost realizat la Universitatea din București, referitor la limba română, în cadrul proiectului DBR-MAT, finanțat de Fundația Volkswagen.

Solutia care a fost cu succes aplicata limbii romane pentru efectuarea analizei sintactice de dependenta, in cadrul proiectului DBR-MAT, este de natura stocastica si se refera la asocierea unei probabilitati fiecarei structuri sintactice, pentru o propozitie data fiind aleasa acea structura sintactica a carei probabilitate asociata are valoarea maxima. Atribuirea unei asemenea probabilitati inseamna gasirea unui model stocastic, si anume a acelui model stocastic care este cel mai adecvat. In aceasta abordare, pentru gasirea structurii sintactice de dependenta a unei propozitii nu este necesara specificarea explicita a unei gramatici de dependenta. Gramatica va fi in mod implicit inclusa in parametrii modelului stocastic, care, la randul lor, vor fi estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru putem spune ca a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii ca in materialul prezentat pe web. Etapele in derularea unui asemenea algoritm sunt: gasirea multimii T ("part of speech tagging"); gasirea multimii P (adica a relatiilor de dependenta); gasirea multimii D (adica a tipului dependentelor). Vom reveni cu detalii asupra modului in care s-a facut gasirea acestor multimi daca sunteti interesat de aspectele stocastice ale acestei abordari.

Va rog sa dati un exemplu de folosire a unui corpus obtinut prin adnotare cu DGA.

Un exemplu de folosire a unui asemenea corpus il constituie efectuarea analizei sintactice (parsing) in maniera stocastica. Grupul nostru a efectuat analiza sintactica de dependenta intr-o abordare stocastica, in care nu este necesara specificarea unei gramatici de dependenta propriu-zise. Gramatica a fost in mod implicit inclusa in parametrii modelului stocastic, care, la randul lor, au fost estimati pe baza datelor lingvistice (adica a unui corpus).

In acest cadru, a gasi un algoritm de analiza sintactica inseamna a gasi un algoritm care are ca input o propozitie si ca output structura sintactica (S,D) a acelei propozitii, unde $S=(T,P)$ si D au aceleasi semnificatii din materialul publicat pe web. Etapele in derularea unui asemenea algoritm sunt: gasirea multimii T ("part of speech tagging"); gasirea multimii P (adica a relatiilor de dependenta); gasirea multimii D (adica a tipului dependentelor).

Gasirea multimii T s-a facut utilizandu-se un algoritm propus de Ratnaparkhi in 1996. Acest algoritm este de natura stocastica si utilizeaza entropia maxima. Gasirea multimii P s-a facut, de asemenea, prin utilizarea unui algoritm stocastic, si anume a algoritmului lui Eisner, propus in acelasi an. Acest algoritm a fost modificat de noi prin schimbarea modelului stocastic, cu utilizarea din nou a entropiei maxime. Algoritmul de gasire a multimii P reprezinta o implementare a metodei programarii dinamice cu scopul de a gasi cea mai probabila analiza in

maniera "bottom-up" (de jos in sus). Dupa determinarea multimilor T si P, gasirea multimii D nu mai ridica nici un fel de probleme.

Programul DGA se bazeaza in totalitate pe utilizator in procesul de adnotare sau reprezinta un sistem semiautomat, care ii ofera utilizatorului niste optiuni din care sa aleaga? Daca nu este semiautomat (asa cum pare), care este motivul pentru care a fost proiectat in acest fel? Chiar daca sistemul porneste fara nici un fel de cunostinte prealabile, in timp el ar putea acumula niste sabloane.

DGA nu este semiautomat, in sensul ca nu are un mecanism intern care sa adnoteze initial un corpus, iar mai apoi acesta sa ii fie prezentat utilizatorului pentru corectare. Aceasta pentru a face DGA cat mai independent de limba si de variantele de formalizare ale gramaticilor de dependenta. Totusi, DGA poate fi usor transformat intr-un instrument semiautomat prin integrarea unor produse externe (POS tagger, parser etc.). DGA permite vizualizarea si modificarea corpusurilor adnotate anterior (folosind comanda Open corpus din meniul File). Desi initial aceasta facilitate a fost prevazuta pentru a modifica adnotari facute tot cu DGA, ea poate fi folosita si in cazul unor produse externe (POS tagger, parser etc.). Trebuie doar ca, corpusul adnotat automat (cu produsul extern) sa fie transformat din formatul pe care il foloseste produsul extern respectiv in formatul XML folosit de DGA. Dupa aceasta operatie corpusul poate fi deschis cu DGA si corectate (modificate) adnotarile facute automat.

Exista posibilitatea de a ajuta procesul de adnotare cu DGA folosind produse externe (de ex. POS tagger, parser etc.)? Utilizatorul in acest caz ar avea rolul de a corecta o adnotare facuta automat si deci procesul de adnotare ar fi mult mai rapid.

Da, exista aceasta posibilitate. DGA permite vizualizarea si modificarea corpusurilor adnotate anterior (folosind comanda Open corpus din meniul File). Desi initial aceasta facilitate a fost prevazuta pentru a modifica adnotari facute tot cu DGA, ea poate fi folosita si in cazul unor produse externe (POS tagger, parser etc.). Trebuie doar ca, corpusul adnotat automat (cu produsul extern) sa fie transformat din formatul pe care il foloseste produsul extern respectiv in formatul XML folosit de DGA. Dupa aceasta operatie corpusul poate fi deschis cu DGA si corectate (modificate) adnotarile facute automat.

Poate fi modificat DGA, in principiu, astfel incat sa permita, in egala masura, adnotarea morfosintactica a textelor? Cum ar putea fi realizat acest lucru? (pusa de doua ori)

Da. Trebuie doar ca DGA sa mai permita si adaugarea informatiilor morfologice pentru un cuvânt. Acest lucru se poate face foarte simplu adaugand in meniul contextual care se deschide când se face clic dreapta pe un cuvânt o comandă "morphology" (de exemplu) care, atunci când este apelată, să deschidă o casetă de dialog unde să fie introduse informațiile morfologice pentru cuvântul respectiv.

Am inteles ca DGA salveaza rezultatele in format XML. Dispuneti si de un XSLT care sa transforme rezultatele din format XML in alt format XML si, daca da, in care anume? Iar daca nu, asta inseamna ca ii revine utilizatorului sarcina de a scrie un XSLT?

Formatul XML folosit de DGA este unul foarte simplu, inspirat din standardul XCES. Nevoile utilizatorilor pot fi însă foarte variate așa că, dacă utilizatorul are nevoie de un alt format, atunci trebuie să scrie un XSLT cu care să transforme corpul din formatul XML folosit de DGA în formatul care îi trebuie. De exemplu, eu folosesc un XSLT pentru a transforma textele aadnotate cu DGA în format HTML care să permită vizualizarea acestor texte pe web.

Cum se pot vizualiza fisierele XML rezultate in urma adnotarii cu DGA online? (pusa de doua ori)

Exista mai multe solutii posibile. Voi prezenta una care a fost deja implementata:

Mai intai, fisierele XML rezultate in urma adnotarii cu DGA au fost transformate cu ajutorul XSLT in fisiere HTML. In fisierele HTML fiecare propozitie este continuta intr-un FORM. La operatia de SUBMIT (click pe o propozitie in cazul nostru), FORM-ul va trimite server-ului (cu ajutorul unor campuri de tip HIDDEN) informatiile cuprinse in adnotare. Pe baza acestor informatii, un script perl de pe server construiește o imagine jpeg care să reprezinte în forma grafică obisnuită adnotarea. Aceasta imagine îi este returnată browser-ului care o va afișa într-o fereastră nouă. Puteti vedea cum functioneaza aceasta solutie la adresa:

<http://phobos.cs.unibuc.ro/roric/texts/indexro.html>

Ce este XCES? (pusa de doua ori)

XCES (XML Corpus Encoding Standard) este un standard de reprezentare a corpurilor. Detalii și alte informații puteți găsi la:

Cum v-ati inspirat din setul de taguri X C E S cand ati proiectat instrumentul DGA? In ce consta asemanarea?

Deoarece nu exista inca un set standard de taguri cu care sa se marcheze adnotarea sintactica a unui text, DGA foloseste pentru reprezentarea textelor adnotate un set de taguri inspirat din XCES (setul de taguri standard pentru reprezentarea adnotarii morfosintactice). Ideea generala a fost aceea de a folosi un set de taguri cat mai simplu pentru a putea fi usor facut compatibil cu un standard viitor. Din XCES s-au pastrat tagurile care marcau structura generala (delimitarea propozitiilor cu `<s>...</s>`, a fiecarui token din cadrul unei propozitii cu `<tok>...</tok>`). Pentru fiecare token, tot din XCES, s-a pastrat marcarea formei ortografice cu `<orth>...</orth>` si a partii de vorbire (neambigua) cu `<ctag>...</ctag>`. S-a renuntat la tagurile (din XCES) care se refereau la informatiile morfologice si s-au introdus taguri noi pentru informatiile sintactice: `<syn>...</syn>`, `<head>...</head>`, `<reltype>...</reltype>`.

Care sunt tipurile de dependente posibile?

Tipurile clasice de dependente sunt: subiect, obiect si complement (altul decat cel direct). Aceste dependente pot fi insa in continuare rafinate. De pilda, in stabilirea celor mai frecvente relatii de dependenta in limba romana s-a luat in considerare, de cele mai multe ori, functia sintactica a cuvintului dependent. Un tabel continand cele mai frecvente tipuri de dependente in limba romana poate fi gasit in articolul aflat acum pe web. Acest tabel va fi actualizat de RORIC la sfarsitul lunii februarie.

Pentru ce anume este relevanta detectarea relatiilor de dependenta?

Detectarea relatiilor de dependenta este relevanta in special deoarece majoritatea lingvistilor sunt astazi de acord cu faptul ca in centrul conceptului de structura a propozitiei se afla relatiile dintre cuvinte, indiferent daca aceste relatii se refera la posibilele functii gramaticale (subiect, complement etc.), ori la acele legaturi care imbina cuvintele in unitati mai largi, cum ar fi grupurile sintactice. Spre deosebire de gramaticile generative, gramaticile de dependenta pot descrie cu mai mult succes fenomene lingvistice cum ar fi existenta constituentilor discontinui sau variatia ordinii cuvintelor in cadrul propozitiei. In ceea ce priveste aplicatiile de natura computationala, formalismul gramaticilor de dependenta s-a dovedit, spre exemplu, a fi mai adecvat decat cel al gramaticilor generative pentru a fi utilizat cu succes in efectuarea analizei sintactice de tip stocastic ("stochastic parsing").

Intrebări privind aplicarea gramaticilor de dependență la limba română

Dati exemple de cateva diferente între relațiile sintactice clasice și relațiile de dependență.

Diferența cea mai importantă este constituită de subordonarea prepozițională. În sistemul relațiilor de dependență, prepoziția își pierde, în general, calitatea de element subordonator și, în consecință, stabilește ea însăși diferite relații. Conventional, prepoziția preia relația cuvântului pe care îl preceda (atribut, complement etc.), iar acesta îi se subordonează printr-o relație numită prepozițională. În alte situații (spre ex. prepoziția A, morfem al infinitivului), prepoziția se subordonează verbului regent nepredicativ (prin relație infinitivală), iar acesta, la rândul său, unui alt regent. De asemenea, din cauza dublei subordonări din sintaxa tradițională, am considerat, prin convenție, elementul predicativ suplimentar ca fiind complement circumstanțial (de mod). O altă diferență privește articolele: articolul nehotărât intră în relație nehotărâtă cu substantivul regent, articolul hotărât (antepus numelor proprii la G/D) stabilește o relație hotărâtă cu regentul, articolul posesiv se subordonează regentului, preluând funcția cuvântului în G. etc.

În unele situații nu există o continuitate între enunțuri. De ce?

Această discontinuitate între fragmentele de text se explică prin adnotarea unor fraze mai ample și prin segmentarea lor în propoziții. În cazul propozițiilor subordonate de diverse feluri, și, mai ales, în cazul celor intercalate, izolarea din contextul în care apar determină o rupere a continuității logice a enunțului.

Care este soluția propusă în situații ambigue din punct de vedere sintactic?

Am încercat să evit texte care să conțină asemenea situații, complicate, de altfel, pentru toate tipurile de gramatici. Totuși, dacă, inevitabil, am fost în situația de a rezolva un exemplu ambiguu, am apelat la argumente contextuale. Într-un exemplu de tipul: Soluția fiind lăsată în suspans, ședința s-a încheiat, contextul ne indică, totuși, o valoare pasivă a participiului, iar dependentele sunt următoarele: FIIND stabilește o relație auxiliară față de LASATA care, la rândul lui, depinde de verbul (s-a) INCHEIAT printr-o relație de complement circumstanțial (temporal).

Cum ati rezolvat problema locutiunilor (de orice fel)?

Am considerat, principal si conventional, ca, din punctul de vedere al relatiilor de dependenta, nu exista locutiuni. Am incercat, in masura posibilului, sa analizez in elemente componente locutiunile intalnite. In acelasi timp m-am straduit sa evit orice locutiune a carei structura nu se poate analiza.

Cum ati rezolvat problema locutiunilor prepozitionale alcatuite dintr-un adverb si o prepozitie? Dati un exemplu.

In exemple de tipul: inainte de, aproape de etc., am considerat doua unitati diferite (adverb si prepozitie), primul fiind cap pentru al doilea. Relatia stabilita de prepozitie este: complement indirect, iar relatia stabilita de adverb fata de un alt cuvânt-cap este, in general: complement circumstantial: A plecat inainte de masa.

Cum ati rezolvat problema numelor compuse?

Daca numele compuse prezinta o clara structura sintactica, ele sunt analizate in elemente componente (intre care se stabilesc relatii de dependenta): Statele Unite, Ministerul de Externe, Marea Britanie; de la, pana la etc. In ceea ce priveste numele proprii compuse, le-am considerat o singura unitate lexicala.

Cum ati rezolvat problema numelor proprii compuse?

Numele proprii compuse (substantive compuse) au fost considerate, conventional, un singur cuvânt. Tehnic, am eliminat spatiul dintre elementele componente.

Cum ati rezolvat notarea diferita a numeralelor gasite in textele adnotate (spre ex., 100 de mii, comparativ cu 100,000)?

In prima situatie (100 de mii), am considerat 100 numeral, de care depinde prepozitia DE printr-o relatie de "atribut substantival"; MII, ca substantiv, depinde de prepozitie prin "relatie prepozitionala". In cazul al doilea, am considerat un simplu si unic numeral cardinal.

Explicati cum ati procedat la diateza pasiva.

Explicatia este urmatoarea: verbele auxiliare intra, toate, in relatie de dependenta cu participiul verbului de conjugat. Aceasta relatie de dependenta se numeste: relatie auxiliara.

Cum se face diferența între coordonarea prin conjuncții și cea prin juxtaponere?

În cazul coordonării prin conjuncții, cele două unități coordonate depind de conjuncția coordonatoare prin relația numită conjuncțională, iar conjuncția se subordonează elementului regent, preluând funcția determinantelor. În cazul coordonării prin juxtaponere, toate elementele subordonate intra în dependența directă cu elementul regent unic.

Cum ați rezolvat problema așa-numitelor "construcții" gerunziale și infinitivale?

Ca și locuțiunile sau expresiile, aceste "construcții" nu au fost considerate ca atare (de altfel, ele sunt, oricum, susceptibile de obiectii). În consecință, am interpretat toate situațiile de acest tip ca structuri analizabile. Iată un exemplu: Fiind plecată din oraș, nu a văzut ce s-a întâmplat. FIIND se subordonează verbului predicativ VAZUT (nu a văzut) prin relație de complement circumstanțial (de cauză), iar PLECATĂ este nume predicativ depinzând de regentul gerunzial (și copulativ) FIIND.

Coordonarea poate fi realizată, simultan, prin virgulă (juxtaponere) și prin conjuncție. Explicați, printr-un exemplu, relațiile de dependență stabilite.

Un exemplu de acest tip poate fi următorul: Colocviul a fost antrenant, interesant și plin de discuții pasionate. Adjectivele ANTRENANT, PASIONANT și PLIN sunt, toate, în gramatica "tradițională", nume predicative față de verbul copulativ A FOST. În cazul relațiilor de dependență, numai ANTRENANT este considerat nume predicativ; celelalte două se subordonează conjuncției coordonatoare SI (printr-o relație "conjuncțională"); conjuncția SI preia, astfel, funcția sintactică a adjectivelor și intra, ea însăși, în relație de "nume predicativ" față de verbul A FOST.

Ați făcut vreo diferență, în cadrul relațiilor de dependență propuse, între gradele comparativ și superlativ?

Nu am făcut nici o diferență între cele două grade de comparație, subordonând toate situațiile din această categorie unei unice relații de dependență, numită generic "relație comparativă". În acest fel, am respectat un anumit grad de rafinare, principal propus în realizarea relațiilor de dependență.

Explicati ce tipuri de relatii de dependenta apar in contextul verbului a avea + verb la supin.

Relatiile care apar sunt urmatoarele: complement direct si relatie prepozitionala. Iata un exemplu minimal: Am de lucrat. Prepozitia DE (marca a supinului) stabileste fata de verbul regent AM relatia de complement direct, iar verbul de conjugat LUCRAT se subordoneaza prepozitiei, devenita, la randul ei, regent, prin relatia numita prepozitionala.

Cum rezolvati, din punctul de vedere al relatiilor de dependenta, un enunt de tipul: Imaginea este stearsa?

Acest enunt, lipsit de orice argument contextual suplimentar, este, intr-adevar, ambiguu din punct de vedere sintactic. Solutia este, in asemenea situatii, arbitrara: fie ESTE verb predicativ (si copulativ), iar STEARSA adjectiv (relatie de "nume predicativ"), fie ESTE verb auxiliar (diateza pasiva) si depinde de regentul sau STEARSA (considerat verb predicativ) (relatie "auxiliara"). In cazul de fata, totusi, inclin spre prima varianta.

Cum ati numi relatiile de dependenta dintr-un context nominal de tipul: (Am admirat) o pictura a lui Picasso?

In acest context, relatiile de dependenta sunt urmatoarele: articolul nehotarat O depinde de substantivul PICTURA printr-o relatie numita "nehotarata"; articolul posesiv A se subordoneaza tot lui PICTURA prin relatia "atribut substantival"; articolul hotarat LUI (antepus datorita vecinatatii unui nume propriu masculin) se subordoneaza acestuia (lui PICASSO) printr-o relatie numita "hotarata"; in sfarsit, substantivul PICASSO depinde de articolul posesiv A printr-o relatie "posesiva".

Exista vreo relatie de dependenta in limba romana pentru care cuvantul-cap si cuvantul dependent sa fie prepozitii? Daca da, oferiti un exemplu! Cum ati numi o astfel de relatie?

Exista, in acest sens, situatia prepozitiilor compuse (in care, prin conventie, al doilea element depinde de primul). Exemplu: Cartea de la tine a fost interesanta. Prepozitia LA depinde de prepozitia DE, iar relatia se numeste PREPOZITIONALA (cuvantul-cap fiind o prepozitie).

Dati doua exemple de noi relatii de dependenta pentru limba romana aparute in urma adnotarii textelor din ziar.

1. relatie prepozitionala, in care intotdeauna cuvantul-cap este o prepozitie, indiferent de elementul subordonat.
2. relatie conjunctiionala, in care intotdeauna cuvantul-cap este o conjunctie coordonatoare, indiferent de elementul subordonat.

Dati un exemplu de propozitie romaneasca care sa contina o relatie prepozitionala in care cuvantul-cap este o prepozitie iar cuvantul dependent este o conjunctie coordonatoare.

Un posibil exemplu este urmatorul: A fost acceptat dupa proba practica si interviu. PROBA si INTERVIU depind de conjunctia coordonatoare SI (relatie conjunctiionala), iar aceasta de prepozitia regenta DUPA (relatie prepozitionala).

Dati un exemplu de propozitie romaneasca care sa contina o relatie prepozitionala in care cuvantul-cap este o prepozitie iar cuvantul dependent este un pronume demonstrativ.

Un posibil exemplu este urmatorul: Putini dintre acestia au acceptat. Prepozitia DINTRE este cuvantul-cap, iar pronumele ACESTIA este cuvantul dependent. Relatia este prepozitionala.

Exista vreo relatie de dependenta in limba romana pentru care cuvantul-cap sa fie articol posesiv, iar cuvantul dependent sa fie un pronume? Cum s-ar numi o astfel de relatie? In cazul in care ea exista dati un exemplu.

Da, exista. Iata un exemplu: Aceste interese ale lui nu ma intereseaza. Articolul posesiv ALE este cuvantul-cap, iar pronumele personal LUI este cuvantul dependent. Relatia se numeste RELATIE POSESIVA.

Ce tipuri de relatii de dependenta ati gasit, pentru limba romana, in care cuvantul-cap sa fie o prepozitie? Dati exemple.

Atunci cand cuvantul-cap este prepozitie, toate relatiile de dependenta (indiferent de termenul dependent) se numesc RELATII PREPOZITIONALE. De o prepozitie pot depinde: un substantiv, un adverb, un pronume, un verb nepredicativ, o conjunctie coordonatoare, un numeral.

Dati un exemplu de relatie de dependenta de tip "nume predicativ" formata cu un cuvânt-cap verb si un cuvânt dependent numeral (in limba romana).

Exemplul este urmatorul: Premiantii din aceasta clasa sunt trei. In acest caz, numeralul cardinal TREI este nume predicativ fata de verbul copulativ SUNT. Relatia se numeste predicativa.

Dati exemplu de un subiect exprimat printr-o conjunctie.

Un exemplu simplu poate fi acesta: Americanii si britanicii au bombardat sistematic Afganistanul. Cele doua substantive depind de conjunctia coordonatoare prin relatie coordonatoare, iar SI este in pozitie de subiect fata de verbul regent AU BOMBARDAT.

Explicati, inclusiv printr-un exemplu, cum arata un nume predicativ "exprimat" printr-o prepozitie.

Iata exemplul: Aceasta replica nu este in masura sa ma ajute. In acest caz, prepozitia IN intra in relatie de nume predicativ cu verbul regent copulativ ESTE (preluand, in realitate, functia sintactica a intregii sintagme IN MASURA). Substantivul MASURA intra in relatie prepozitionala fata de elementul regent IN.

Relatia auxiliara priveste exclusiv verbele auxiliare?

Nu. Sunt doua tipuri de relatie auxiliara: primul se realizeaza intre elementul regent verbal si verbele auxiliare (la moduri si timpuri compuse, ca si la diateza pasiva); al doilea se realizeaza intre elementul regent verbal si conjunctia SA, element component si morfem specific al conjunctivului. In ambele situatii, verbul este cuvântul-cap, iar verbele auxiliare, respectiv conjunctia auxiliara sunt cuvinte dependente.

Dati un exemplu de relatie de dependenta in limba romana pentru care cuvântul-cap sa fie un numeral, iar cuvântul dependent sa fie substantiv, in cazul in care o astfel de relatie exista.

Acest tip de relatie exista si se numeste: atribut substantival. Iata un exemplu: Pe 11 septembrie a avut loc un atentat asupra Statelor Unite.

Puteti da un exemplu de propozitie in limba romana in care sa intervina o relatie de dependenta in cadrul careia cuvantul-cap este un articol posesiv, iar cuvantul dependent este un adjectiv? Cum ati numi o astfel de relatie de dependenta?

Exemplul este urmatorul: M-a vizitat un prieten al meu. Relatia se numeste **POSESIVA**.

Dati un exemplu de relatie de dependenta, in limba romana, in care cuvantul-cap sa fie adjectiv, iar cuvantul dependent un articol demonstrativ.

Un exemplu poate fi urmatorul: Cea mai frumoasa casa este a ta. Relatia se numeste: relatie comparativa.

Un exemplu poate fi urmatorul: Cea mai frumoasa casa este a ta. Relatia se numeste: relatie comparativa. (pusa de doua ori)

Sunt, pana acum, cinci tipuri. Iata relatiile rezultate:

1. Complement direct: L-a intrebat pe colegul ei daca a invatat.
2. Complement indirect: Se gandeste la vacanta.
3. Complement de agent: A fost ajutat de tatal lui.
4. Nume predicativ: Astazi sportivul nu a fost in forma.
5. Complement circumstantial: Se plimba cu masina.

Dati un exemplu de relatie de dependenta, in limba romana, in care cuvantul-cap sa fie un adjectiv, iar cuvantul dependent o prepozitie.

Cunosc o persoana capabila de orice. **CAPABILA** este cuvantul-cap, **DE** este cuvantul dependent, iar relatia este: complement indirect.

Dati exemple de situatii in care o conjunctie este cuvantul-cap si explicati procedeu.

Un exemplu poate fi urmatorul: Am primit o carte interesanta si utila. Adjectivele **INTERESANTA** si **UTILA** se subordoneaza regentului **SI** (devenit astfel cuvantul-cap) prin relatii conjunctionale. La randul sau, conjunctia coordonatoare **SI** se subordoneaza substantivului **CARTE** prin relatia de atribut adjectival (preluand functia traditionala a acestora). Un alt exemplu: Sunt interesat de fizica si de matematica. Prepozitiile **DE** se subordoneaza amandoua cuvantului-cap **SI** prin relatii conjunctionale. **SI** se subordoneaza verbului prin relatia de complement

indirect. Cele doua substantive (FIZICA si MATEMATICA) se subordoneaza prepozitiilor DE prin relatii prepozitionale.

In sfarsit, un alt exemplu: Relatiile de prietenie si colegialitate sunt preferabile celor de dusmanie si invidie. PRIETENIE si COLEGIALITATE se subordoneaza cuvantului-cap SI prin relatii conjunctionale, conjunctia SI se subordoneaza cuvantului-cap DE printr-o relatie prepozitionala, iar acesta substantivului regent RELATIILE prin relatia de atribut substantival.

Explicati, pe scurt, care este rolul prepozitiei in sistemul relatiilor de dependenta.

Prepozitia are urmatorul rol: in pozitie de cuvant dependent, preia functia sintactica a determinantului (complement, atribut, nume predicativ etc.); acesta din urma se subordoneaza, la randul lui, prepozitiei si stabileste, fata de aceasta, intotdeauna o relatie prepozitionala.

Numele predicativ poate fi exprimat printr-o prepozitie?

Da. Iata un exemplu: Acest costum iti este pe masura. Prepozitia PE este nume predicativ, subordonat verbului regent copulativ ESTE.

Exista vreo relatie de dependenta in limba romana pentru care cuvantul "cap" sa fie o prepozitie, iar cuvantul "dependent" sa fie un pronume? Daca da, cum s-ar numi o astfel de relatie? In cazul in care ea exista, dati un exemplu.

Da, exista. Un exemplu poate fi urmatorul: Multi dintre noi ar vrea sa vada Italia. Prepozitiei DINTRE (cuvant-cap) i se subordoneaza pronumele personal NOI prin asa-numita relatie prepozitionala (care apare intotdeauna cand cuvantul-cap este o prepozitie).

Care este diferenta dintre "conjunctie auxiliara" si "conjunctie coordonatoare"?

Diferenta este urmatoarea: conjunctia coordonatoare (SI, SAU, ORI) coordoneaza doua parti de propozitie de acelasi fel, iar cea auxiliara este reprezentata de morfemul de conjunctiv SA. Prima preia functia elementelor coordonate, a doua se subordoneaza, ea insasi, verbului la conjunctiv printr-o relatie auxiliara.

Relatia auxiliara se refera strict la verbele auxiliare sau nu?

Nu.Sunt doua posibilitati: prima priveste verbele auxiliare (care intra in relatie de dependenta cu verbul de conjugat); a doua priveste conjunctia auxiliara SA (morfem al conjunctivului). Ambele relatii se numesc auxiliare. Ex: As fi vrut sa vin acasa mai repede. AS si FI sunt verbe auxiliare (fata de VRUT), iar SA este conjunctie auxiliara (fata de VIN).

Dati un exemplu de relatie de dependenta in limba romana, in care cuvantul "cap" sa fie o prepozitie, iar cuvantul "dependent" sa fie o conjunctie.

Un exemplu este urmatorul: Colaborarea dintre SUA si Anglia este remarcabila. Cuvantul "cap" este DINTRE, cuvantul "dependent" este SI. Relatia este prepozitionala. La randul lor, substantivele SUA si ANGLIA depind de SI prin relatii conjunctionale. Prepozitia DINTRE se subordoneaza substantivului regent prin functia (relatia) de atribut substantival.

Dati un exemplu de relatie de dependenta in limba romana pentru care cuvantul-cap sa fie un verb nepredicativ, iar cuvantul dependent sa fie un numeral, in cazul in care o astfel de relatie exista.

Exista. Iata un posibil exemplu: Fiind al doilea din clasa, dupa decernarea premiilor, el a ramas cam suparat.FIIND este verbul nepredicativ, AL DOILEA este numeral ordinal, iar relatia este de nume predicativ.

Exista vreo relatie de dependenta in limba romana pentru care cuvantul-cap sa fie o prepozitie, iar cuvantul dependent sa fie un numeral si cum s-ar numi o astfel de relatie? In cazul in care ea exista, dati un exemplu.

Exista. Un posibil exemplu este acesta: Numai doi concurenti din zece au terminat cursa. Prepozitia DIN este cuvant-cap pentru numeralul cardinal ZECE. Relatia de dependenta stabilita intre prepozitie (cap) si numeral (dependent) se numeste PREPOZITIONALA.

Exista vreo relatie de dependenta in limba romana pentru care cuvantul-cap sa fie o prepozitie, iar cuvantul dependent sa fie un verb predicativ si cum s-ar numi o astfel de relatie? In cazul in care ea exista, oferiti un exemplu.

Nu exista. Este imposibila, din punctul de vedere al logicii sistemului relatiilor de dependenta.

Exista vreo relatie de dependenta in limba romana pentru care cuvantul-cap sa fie un pronume, iar cuvantul-dependent sa fie o prepozitie? Cum s-ar numi o astfel de relatie? In cazul in care ea exista, dati un exemplu de propozitie romaneasca in care sa intervina.

Exista. Iata un exemplu: Unii dintre colegi nu au venit astazi la facultate. Pronumele nehotarat regent (cap) este UNII, iar DINTRE este prepozitia subordonata. Relatia se numeste atribut substantival.

Exista vreo relatie de dependenta in limba romana pentru care cuvantul "cap" sa fie un adjectiv, iar cuvantul "dependent" sa fie o prepozitie? Cum s-ar numi o astfel de relatie? In cazul in care exista, dati un exemplu de propozitie in romana in care sa intervina.

Exista. Iata un exemplu: Bucuros de vestea primita, a plecat acasa. Adjectivul BUCUROS este "cap", iar prepozitia DE este "dependent". Relatia este: complement indirect. Un alt exemplu de relatie poate fi complement circumstantial: Odata plecati din oras, nu s-au mai intors.

Explicati printr-un exemplu relatia hotarata.

Relatia hotarata este posibila numai in cazul numelor proprii in genitiv si in dativ (masculine si, in unele situatii, feminine), precedate de articolul hotarat LUI. Iata un exemplu: I-am cerut scuze lui Petru. LUI se subordoneaza lui PETRU prin relatia hotarata. PETRU se subordoneaza lui AM CERUT prin relatia de complement indirect.

Explicati relatiile in cadrul prepozitiilor compuse.

In cadrul prepozitiilor compuse, am considerat, conventional, primul component pe post de cuvant-cap (care depinde, la randul lui, de un alt element regent); al doilea component se subordoneaza primului prin relatie prepozitionala; in sfarsit, cuvantul pe care il introduce prepozitia compusa respectiva se subordoneaza, de asemenea, primului component, prin aceeasi relatie prepozitionala. Ex: Vin de la scoala. DE este cuvant-cap pentru LA si pentru SCOALA. El se subordoneaza verbului prin relatia de complement circumstantial (preluand rolul intregului element de relatie subordonator).

Explicati diferenta dintre relatiile posesiva si nehotarata (prin raportarea la cuvantul-cap).

In cazul relatiei posesive, articolul posesiv devine cuvânt-cap (preia rolul substantivului regent, cu care se acorda formal: carte a colegului); substantivul in genitiv se subordoneaza lui A, iar relatia este posesiva. In cazul relatiei nehotarate, articolul nehotarat se subordoneaza substantivului pe care il preceda si de care depinde formal (cumpar o carte); este, asadar, un element dependent.

De ce, dintre toate tipurile de pronume din limba romana, ati particularizat pronumele reflexiv?

Explicatia este urmatoarea: pronumele reflexiv, dintre toate celelalte categorii, este singurul cu valoare morfematica (ajuta la formarea diatezei reflexive). Pentru a evidientia aceasta diateza, am recurs la relatia reflexiva (realizata cu ajutorul pronumelui reflexiv).

In situatia in care cuvantul cap este o prepozitie, relatia de dependenta corespunzatoare va fi intotdeauna numita relatie prepozitionala? (pusa de doua ori)

In toate situatiile de acest gen (cuvantul-cap prepozitie), relatia va fi numita prepozitionala.

Care sunt elementele morfologice care intra in relatie de dependenta cu o conjunctie coordonatoare? Dati cateva exemple.

Conjunctia coordonatoare poate fi cuvânt-cap in situatii de felul urmatoar:

1. fata de substantive: Am vizitat Roma si Milano.
2. fata de pronume: Nu stiu daca au intrat multi sau putini.
3. fata de adjective: Ea a devenit frumoasa si inteligenta.
4. fata de numeral: Au ramas doar doi sau trei dintre noi.
5. fata de prepozitii: S-a plimbat prin Iasi si prin Bucuresti.

Exista vreo relatie de dependenta in limba romana pentru care cuvantul-cap sa fie un verb, iar cuvantul dependent sa fie o prepozitie? Daca da, cum s-ar numi o astfel de relatie? In cazul in care ea exista, dati un exemplu.

In limba romana exista numeroase asemenea situatii. Denumirea relatiei difera in functie de rolul sintactic al cuvântului precedat de prepozitia respectiva: daca

prepozitia preceda un nume predicativ, relatia e predicativa, daca preceda un complement circumstantial, relatia e circumstantiala, daca preceda un complement direct, relatia e de complement direct etc. Iata un posibil exemplu: Plec la facultate. In acest context, LA stabileste o relatie de complement circumstantial fata de verbul regent PLEC, iar substantivul FACULTATE intra intr-o relatie prepozitionala fata de cuvantul-cap LA.

Dati cate un exemplu de propozitie romaneasca in care sa intervina cate o relatie de tip "nume predicativ", formata cu un cuvant-cap verb si un cuvant dependent adverb, respectiv pronume.

Exemple pot fi urmatoarele:

1. cuvant-cap verb, dependent adverb: Este bine in vacanta.
2. cuvant-cap verb, dependent pronume: Intrebarea este aceasta.

In ambele situatii, verbul regent este copulativ, iar relatia este "nume predicativ".

Intrebări referitoare la HPSG

Cine a inventat HPSG ?

Fondatorii HPSG sînt Ivan A. Sag (profesor de lingvistică și sisteme simbolice la Universitatea Stanford California) și Carl Pollard (profesor de lingvistică la Universitatea statului Ohio, Columbus).

Există la universitate cursuri de introducere în HPSG?

Până anul trecut a existat un curs introductiv pentru ultimul an și un curs cu aplicații la limba română pentru anul de masterat - ambele la Facultatea de Litere de la Universitatea din București. Prin decizia șefului catedrei de limba română, din acest an nu mai există decît cursul de la masterat.

Există și alte prezentări în limba română ale HPSG?

Există, de asemenea, o prezentare sintetică în Doina Tatar, "Inteligența artificială", Editura Albastru, Cluj, 2001.

Cum as putea gasi mai multe lucrari de HPSG? (pusa de doua ori)

Formeaza hpsg drept cuvânt-cheie si vei gasi o adresa pentru "HPSG literature", care contine o foarte bogata bibliografie.

Exista implementari computationale ale HPSG?

Da, exista. Cele mai recente (cunoscute de noi) sunt implementarile gramaticii interogativelor în engleza. De asemenea, exista o implementare referitoare la ordinea cuvintelor în germana.

Au fost deja implementate mecanisme computationale bazate pe HPSG?

Nu sunt sigur ca înțeleg corect ceea ce înțelegeți prin "mecanisme computazionale bazate pe HPSG". Daca aveti în vedere principii HPSG (de exemplu, principiul sintactic al trasaturilor de centru, care pentru mine chiar este un "mecanism computational bazat pe HPSG"), atunci v-as raspunde ca nu cunosc vreo lucrare care sa se fi ocupat în mod concret cu asa ceva. Cu toate acestea, cel care doreste sa implementeze un fragment al unei gramatici, nu poate evita implementarea principiilor foarte generale. Prin urmare, chiar daca nu sunt capabil sa indic o lucrare tratand în mod specific acest subiect, as spune ca astfel de implementari chiar trebuie sa existe si ca implementarea principiilor la care m-am referit este fezabila.

Ce utilitate poate avea implementarea computationala a unei analize HPSG?

Principala utilitate consta în faptul ca analiza devine testabila si poate oferi ipoteze privitoare la plauzibilitatea psihologica a modului în care se achizitioneaza structura supusa analizei.

Ce limbaj de programare se foloseste pentru aplicatiile computationale ale HPSG?

Noi cunoastem aplicatii în PROLOG, dar credem ca exista si aplicatii în LISP.

Ce limbaje de programare recomandati pentru aplicatiile computazionale ale HPSG?

Majoritatea aplicatiilor sunt în PROLOG (în special în Europa). Aceasta, fireste, nu înseamna subestimarea celui alt limbaj, LISP.

Se poate asimila conceptul de unificare celui de reuniune din teoria multimilor?

Da, se poate. Ceea ce se unifica poate fi privit și ca o reuniune.

Ce fenomene lingvistice se modelează cel mai bine prin intermediul formalismului HPSG ?

HPSG se dorește a fi o gramatică a unei limbi în general. Nu există, de aceea, fenomene preferate și fenomene în dizgratie. Se poate însă spune că, pentru analiza dependentelor la distanță (precum "Who do you think killed Kennedy ?"), s-a lucrat mai multă vreme, primele rezultate nefiind satisfăcătoare. Analiza propozițiilor relative a beneficiat, în mod special, de reconsiderări succesive.

Este HPSG o gramatică universală?

Radacinile ultime ale HPSG se găsesc în programul lui Chomsky al Gramaticii Universale. În acest sens, HPSG este o versiune a Gramaticii Universale deoarece este în mod firesc interesată de invariantele limbajului uman. Spre deosebire de programul lui Chomsky, însă, HPSG nu privilegiază invariantele. Dimpotrivă, HPSG înțelege să se apropie cu egal interes de complexitatea și bogăția idiomatice a limbilor concrete. În acest sens, teoria este mai degrabă "tradițională".

Cum se comportă HPSG față de conceptul de "movement", al lui Chomsky?

HPSG nu folosește operația de deplasare din gramaticile lui Chomsky, deoarece nu consideră că există dovezi convingătoare că această operație există cu adevărat, ca parte a "gramaticii mentale".

De ce refuză HPSG conceptul lui Chomsky de "movement"?

Pentru că nu găsește nici o dovadă empirică pentru această operație.

Împrumută HPSG ceva din arhitectura modulară a computerelor? (pusă de trei ori)

Da, împrumută. Felul în care constrangerile din HPSG sunt verificate seamănă mult cu modul în care ne utilizăm computerele. De exemplu, nu e nevoie să întrerupem programul în care lucrăm dacă vrem să ascultăm un CD la computer. Amandouă lucrurile pot fi făcute fără ca unul să depindă de celălalt. Într-un mod

asemanator, constrangerile pe o structura lingvistica data sunt verificate în mod independent, adica verificarea unei constrangeri nu presupune verificarea alteia.

Ce inseamna lexicalismul strict si este el tipic pentru abordarea HPSG referitoare la teoria gramaticala?

Lexicalismul strict (sau tare) este o optiune de organizare a unei teorii gramaticale, în conformitate cu care structura interna a cuvintelor este independenta de felul în care cuvintele contribuie la articularea sintagmelor. Aceasta optiune nu caracterizeaza numai HPSG. De pilda, tot strict lexicalist este si Programul Minimalist, în ciuda faptului ca versiunea precedenta a programului lui Chomsky - Teoria GB - nu era.

Cum este organizata informatia lexicala în HPSG?

Principalele nivele ale informatiei lexicale în HPSG sunt cel gramatical, cel semantic, cel fonologic si cel pragmatic. Exista, de asemenea, o trasatura care este raspunzatoare de plasamentul cuvântului în sintagma. Ceea ce este specific reprezentarilor lexicale în HPSG este faptul ca sunt bogate - daca sunt comparate, de pilda, cu reprezentarile lexicale folosite în GB. Se întâmpla asa deoarece reprezentarile lexicale sunt raspunzatoare de fenomene precum dependenta la distanta, cuantificare sau anaforicitate, care în alte teorii gramaticale sunt considerate autonome si independente.

Constrangerile prin exceptare si regulile lexicale nu sunt mecanisme declarative si procedurale. E corect sa spun ca HPSG nu este pur declarativa? (pusa de doua ori)

Da, este corect sa spuneti asta. Numai ca în absenta acestor mecanisme nedeclarative, analiza limbii naturale ar fi mult mai dificila.

Care este justificarea valorilor coindexate? (pusa de doua ori)

Valorile coindexate reprezinta un instrument util pentru a semnala identitatea nonaccidentala de informatie lingvistica, în contradistinctie cu identitati accidentale de informatie. De pilda, este esential sa semnalăm ca în propozitia "Pe Ion nimeni nu stie unde sa-l mai caute" grupul nominal "pe Ion" trebuie sa se refere la acelasi individ ca si pronumele neaccentuat I. Daca nu se semnaleaza ca aceasta identitate de informatie este esentiala, exista posibilitatea sa se interpreteze propozitia si în sensul ca pronumele neaccentuat se refera la o persoana diferita de cea indicata prin grupul nominal "pe Ion". Caz în care putem avea si propozitia "Pe Ion nimeni nu stie unde sa-l mai caute pe Vasile". Dimpotriva, în acelasi enunt, pronumele

neaccentuat si pronumele negativ au aceeasi persoana si acelasi numar, dar aceasta identitate de informatie nu este esentiala pentru corectitudinea enuntului. Într-adevar, pronumele negativ poate fi înlocuit cu un grup nominal de o persoana sau de un numar diferit, propozitia ramanand mai departe corecta: "Pe Ion voi nu stiti unde sa-l mai cautati". Aceasta din urma identitate de informatie nu va fi notata prin coindexare.

Cum sunt tratate dependentele la distanta în HPSG?

În mod esential, e vorba de trei constrangeri: una reglementeaza "colectarea" de catre un element lexical a informatiei ca un constituent lipseste. Una reglementeaza transmiterea acestei informatii în sintagmele proiectate de centrul lexical în speta, iar ultima "închide" dependenta, furnizand un constituent care aduce informatia-lipsa. De pilda, pentru propozitia:

(1) Bagels ₁, John always said that he likes ₁.

elementele lexicale **likes** si **said** colecteaza informatia ca lipseste un constituent:

(i) likes ₁

(iv) said that he likes ₁.

Aceasta informatie este transmisa sintagmelor proiectate de likes si respectiv de said:

(i) likes ₁

(ii) he likes ₁

(iii) that he likes ₁

(iv) said that he likes ₁.

(v) always said that he likes ₁

(vi) John always said that he likes ₁.

Structura (vi) se închide prin completarea "golului" cu constituentul relevant: **bagels**.

Cum pot fi clasificate, conform teoriei HPSG, fenomenele legate de acord?

Pollard si Sag ("Head-driven Phrase Structure Grammar", Chicago University Press, 1994, 73-88), clasifica acordul în functie de elementele care realizeaza aceasta relatie:

1. Pronume-antecedent
2. Subiect-verb

3. Determinator-nume

Accentuam faptul ca aceasta clasificare nu este determinata de teoria HPSG ci de limba supusa investigatiei - aici engleza. Daca se lucreaza pe româna, clasificarea se modifica, deoarece româna - limba cu morfologie bogata - face uz si de alte forme de acord în comparatie cu engleza: de pilda, acordul nume-adjectiv.

Cum trateaza teoria HPSG fenomenele legate de acord?

Pollard si Sag , trateaza acordul în functie de elementele care realizeaza aceasta relatie:

1. Pronume-antecedent
2. Subiect-verb
3. Determinator-nume

Accentuam ca aceasta clasificare nu este determinata de teoria HPSG ci de limba supusa investigatiei - aici engleza. Daca se lucreaza pe româna, clasificarea se modifica, deoarece româna - limba cu morfologie bogata - face uz si de alte forme de acord în compartie cu engleza: de pilda, acordul nume-adjectiv.

Va rog descrieti pe scurt felul în care sunt tratate sintagmele în HPSG.

Elementul esential în tratamentul HPSG al sintagmelor este valoarea nonvida pentru trasatura DAUGHTERS (RAMURI). Aceasta înseamna ca o sintagma este obligata sa aiba structura interna în sensul ca ea poate fi descompusa în alte elemente constitutive, care pot fi cuvinte sau de asemenea sintagme, dar nu morfeme. Aceasta este de fapt cea mai generala proprietate a sintagmelor. Ce se întâmpla mai departe si cum sunt ele tratate depinde de limba care este analizata. De pilda, româna, dar nu si engleza, detine o sintagma de tipul centru-marcator, prin intermediul careia sunt puse în evidenta anumite grupuri nominale, care sunt obiect direct:

Ion o iubeste pe Maria
John loves Mary

Intentionati sa aplicati teoria HPSG in viitor pentru limba romana?

O aplicam deja! Vetii putea vedea acest lucru chiar în aceasta pagina, peste cateva luni. Exista de fapt cativa cercetatori care analizeaza limba româna în perspectiva HPSG.

Exista descrieri HPSG ale limbii romane?

Da, exista. Dintre lucrarile publicate în strainatate amintim, în primul rand, analizele Paolei Monachesi privind pronumele neaccentuate. O analiza a negatiei multiple apartinand lui Emil Ionescu a fost publicata în "Proceedings of Formal Grammar Conference", Utrecht, 1999. Este sub tipar o analiza a ordinii constituentilor în grupul nominal (Ana Maria Barbu). Exista apoi un numar de lucrari de licenta nepublicate, dar care dovedesc interesul studentilor pentru aplicatiile HPSG.

Analizati în HPSG propozitia: "Fata rade fericita".

Aceasta propozitie este o sintagma de tipul head-subject. Centrul este el însusi o sintagma (rade fericita), iar subiectul este numele "fata". Sintagma "rade fericita" este de tipul head-adjunct, unde headul este verbul, iar adjunctul este adjectivul "fericita". Acordul acestui adjunct cu subiectul sintagmei "rade fericita" se noteaza prin coindexare: ceea ce este subiect pentru sintagma "rade fericita" este în acelasi timp subiect al adjectivului "fericita".

Cum se poate analiza în HPSG propozitia "Luna straluceste vesela"?

Aceasta propozitie este analizata drept o sintagma de tipul centru subiect (centrul fiind sintagma "straluceste vesela", iar subiectul fiind "luna"). Sintagma "straluceste vesela", la rândul ei, este de tipul centru-adjunct si se descompune în centrul verbal "straluceste" si în adjunctul "vesela". Dubla dependentă a adjectivului "vesela" - pe de-o parte dependent de verbul-centru "straluceste", pe de alta dependent de substantivul-subiect "luna" - este simplu consemnata prin precizarea ca subiectul adjectivului este identic cu subiectul verbului. Procedura face inutila analiza unei asemenea structuri prin derivarea ei, în stilul teoriei transformationale standard, dintr-o structura de baza de tipul "Luna straluceste si este vesela".

Presupunem ca analizam o limba necunoscută prin intermediul unui dicționar frazeologic (fiecărei fraze din limba respectivă îi este asociată o frază în română). Există vreun mod de a descoperi care sunt categoriile gramaticale relevante, care vor constitui matricile de trasaturi? Exemplu: Considerăm ca limba în cauză este engleză, pentru care acordul adj. subs. este nespecificat pentru gen, nr.. Se poate imagina o procedură de scriere a matricilor de trasaturi pentru adjectiv în engleză plecând de la matricea de trasaturi a adj. în română, dicționarul de mai sus și principiile generale ale HPSG?

Nu, nu văd cum așa ceva ar fi posibil. Faptele gramaticale relevante pentru o expresie dintr-o anumită limba nu pot fi deduse din sensul expresiilor în cauză - sens captat prin traducere - împreună cu particularitățile gramaticale ale expresiei care serveste drept "metalimbaj".

II

GENERAREA SEMIAUTOMATĂ A SYNSET-URILOR ȘI CLUSTER-ELOR ROMÂNEȘTI DE TIP WORDNET

ASUPRA GENERĂRII SEMIAUTOMATE A SYNSET-URILOR ȘI CLUSTER-ELOR DE TIP WORDNET CU SPECIALĂ REFERIRE LA LIMBA ROMÂNĂ

Florentina Hristea

1 Introducere. Ce este WordNet

WordNet reprezintă în primul rând o *bază de date lexicală interactivă*, dezvoltată în ultimii 15 ani, pentru limba engleză, la Universitatea Princeton, de către un grup de cercetători condus de profesorul George Miller. În același timp, WordNet poate fi privită ca un *dicționar semantic*, deoarece cuvintele sunt localizate pe baza *afinităților conceptuale* cu alte cuvinte, spre deosebire de cazul dicționarilor clasice, unde cuvintele sunt ordonate alfabetic. Deși este similară unui tezaur, WordNet este mult mai utilă aplicațiilor inteligenței artificiale, întrucât este înzestrată cu o bogată mulțime de relații între cuvinte și sensuri ale cuvintelor.

WordNet conține majoritatea substantivelor, verbelor, adjectivelor și adverbilor limbii engleze, organizate în mulțimi de sinonime numite *synset-uri*. Fiecare *synset* reprezintă un *concept*.

Prin urmare, spre deosebire de dicționarele alfabetice standard, care organizează vocabularul folosind similarități morfologice, WordNet structurează informația lexicală în termeni de sensuri ale cuvintelor. WordNet face corespondența dintre formele tip ale cuvintelor și sensurile acestora utilizând categoria sintactică ca parametru. Astfel, cuvintele aparținând aceleiași categorii sintactice care pot fi folosite pentru a exprima același înțeles sunt grupate într-un același *synset*. Cuvintele polisemantice aparțin mai multor *synset-uri*. Spre exemplu, cuvântul englesc *computer* are două sensuri definite în WordNet, ceea ce face ca el să aparțină la două *synset-uri* diferite, după cum urmează:

(1) {computer, data processor, electronic computer, information processing system}

(2) {calculator, reckoner, figurer, estimator, computer}.

În versiunea sa curentă (versiunea 1.6), WordNet conține 129509 cuvinte organizate în 99643 synset-uri, rețeaua utilizând un număr de 229152 noduri. Cuvintele și conceptele sunt legate între ele prin *relații semantice*. Există în total 299711 asemenea relații. Toate aceste numere sunt însă aproximative, întrucât WordNet continuă să crească. Versiunea 1.7 este acum accesibilă în egala măsură, la adresa:

<http://www.cogsci.princeton.edu/~wn/obtain/>

Relațiile semantice se stabilesc între cuvinte, între cuvinte și synset-uri, precum și între synset-uri. Fiecare cuvânt țintește către unul sau mai multe synset-uri, fiecare dintre acestea corespunzând unui anumit sens al cuvântului respectiv. Prin urmare, diferite cuvinte pot ținti către un același sens (synset). Bogăția mulțimii de relații stabilite între synset-uri este ceea ce face ca rețeaua semantică WordNet să fie atât de puternică și de interesantă pentru diverse tipuri de aplicații. Exemple de relații semantice existente în WordNet sunt **sinonimia** (*synonymy*), folosită pentru a forma synset-urile, **hiperonimia** (*hypernymy*) și **hiponimia** (*hyponymy*), corespunzând relației de tip *isa* și respectiv relației inverse (*reverse isa*), **meronimia** (*meronymy*), corespunzând relației *parte-din*, relația **cauzală** referitoare la verbe și altele. O importanță deosebită este atașată relațiilor de hiperonimie și de hiponimie ca relații între synset-uri.

Cu ajutorul relației de hiperonimie (sau de tip *isa*) conceptele de *substantiv* și de *verb* sunt structurate sub formă de *ierarhii*. Cele de *adjectiv* și de *adverb* au o structură diferită (*cluster*). În WordNet există 11 ierarhii substantivale și 512 ierarhii verbale. Semantica relației de tip *isa* permite unui concept să moștenească toate proprietățile hiperonimelor sale. În plus, proprietățile tipice ale unui concept sunt enunțate sub formă de glosă atașată fiecărui concept în parte. Fiecare glosă include o definiție, una sau mai multe explicații suplimentare și unul sau mai multe exemple.

WordNet reprezintă o bază de date lexicală a limbii engleze care a fost adoptată pe scară largă pentru o întreagă varietate de *aplicații practice* din domeniul inteligenței artificiale și în special din cel al procesării limbajului natural. Mulți cercetători care utilizează WordNet, în special în domeniul inteligenței artificiale, consideră că aceasta reprezintă o bază de cunoștințe lexicală și o valorifică ca atare. *Procesarea cunoștințelor* a dobândit noi dimensiuni în S.U.A. datorită existenței WordNet. În același timp, comunitatea științifică internațională se arată extrem de interesată de dezvoltarea unor baze de date lexicale de tip WordNet pentru cât mai multe limbi, în încercarea de a crea o *infrastructură ontologică uniformă*. Astfel, întrucât mulțimea de bază a relațiilor care leagă între ele conceptele rămâne aceeași, indiferent de limbă, *algoritmii de inferență* pentru extragerea informației pot rămâne aceiași.

Posibilele aplicații ale WordNet în cele mai variate domenii (regăsirea informației, extragerea informației, dezambiguizarea, generarea limbajului natural, învățarea, dicționarele electronice, achiziția de cunoștințe s.a.) sunt citate în peste 300 de lucrări științifice. În ultimii ani a apărut și interesul pentru efectuarea de inferență statistică pe baza WordNet.

Este de menționat faptul că, la mijlocul anilor '90, datorită multiplexelor aplicații dezvoltate pe baza WordNet, a fost puternic resimțită nevoia de a se crea baze de date asemănătoare și pentru alte limbi, în special pentru cele europene. Un imens efort științific și financiar a fost lansat în Europa Occidentală, pentru a se crea așa-numita **EuroWordNet**, utilizând varianta americană WordNet ca model. Acest efort științific s-a concretizat în anul 1996, în cadrul proiectului de cercetare - dezvoltare "EuroWordNet", sub conducerea Universității din Amsterdam:

<http://www.hum.uva.nl/~ewn/>

În prezent există câte o bază de date lexicală de tip WordNet pentru limbile daneză, italiană și spaniolă (fiecare aflată în continuă îmbunătățire) și se lucrează la unele similare pentru limbile germană, franceză și estoniană. Tot în prezent se pune problema creării unor astfel de baze de date lexicale interactive pentru limbile din Europa Centrală și de Est, folosindu-se varianta WordNet a

limbii engleze ca model și adaptând-o specificului fiecărei limbi în parte. Proiectul **BalkanNet**, finanțat de Comisia Europeană, se ocupă în prezent de aceste limbi:

<http://www.ceid.upatras.gr/Balkanet/>.

Eforturile cercetătorilor (informaticienilor) se concentrează și asupra problemei generării automate a unor baze de date de tip WordNet corespunzătoare diverselor limbi, generare care să pornească de la rețeaua semantică WordNet a limbii engleze. În cazul limbii române acest studiu a fost realizat, referitor la substantivele și adjectivele românești, de către echipa RORIC-LING de la Universitatea din București, în cadrul proiectului BALRIC-LING (finanțat tot de Comisia Europeană) și este descris în pagina de web a acestui proiect. Prezentăm, în continuare, algoritmi folosiți pentru generarea semiautomată a synset-urilor și cluster-elor de tip WordNet, cu specială referire la limba română.

2 Algoritmul de traducere

Algoritmul de traducere a unui synset englezesc dat în synset-ul corespunzător dintr-o limbă diferită de limba engleză va folosi așa-numite “mulțimi elementare” sau **e-mulțimi**, concept introdus în [Nikolov și Petrova, 00]. O e-mulțime corespunde sensului unui cuvânt și poate fi definită după cum urmează:

Definiția 2.1

O e-mulțime relativă la un cuvânt este mulțimea sinonimelor corespunzând unui anumit sens al aceluși cuvânt.

Să notăm prin **CE** orice cuvânt englezesc și prin **CS** orice cuvânt străin, adică orice cuvânt al unei alte limbi decât limba engleză. Fie *cuvante* din secvența următoare un CE, în timp ce *cuvants1*, *cuvants2* și *cuvants3* reprezintă echivalentele lui obținute prin traducere, în urma folosirii dicționarului bilingv adecvat:

cuvante cuvants1; cuvants2, cuvants3

Pentru a face distincția între *cuvants1*, *cuvants2* și *cuvants3*, în dicționarele standard se folosesc doi separatori. Astfel, simbolul punct și virgulă separă sensuri diferite ale unui cuvânt dat. La rândul ei, virgula separă acele sinonime care se referă la un același sens al cuvântului. (În cazul de față, *cuvants2* și *cuvants3* sunt sinonime). Această formă a unui dicționar bilingv va fi cea folosită de programele care implementează algoritmul de traducere propus. În exemplul anterior, e-mulțimile care intervin sunt următoarele:

{*cuvants1*} și {*cuvants2*, *cuvants3*}.

Programele care implementează algoritmul de traducere vor genera lista tuturor e-mulțimilor de cuvinte străine corespunzătoare sensului tuturor cuvintelor englezești care intervin într-un synset englezesc dat. Synset-ul străin corespunzător celui englezesc studiat este format din una sau mai multe dintre e-mulțimile generate (care pot fi reunite). “Candidați” la includerea în synset-ul străin sunt *e-mulțimi etichetate*, adică acele e-mulțimi care conțin *cuvinte etichetate*.

Pentru a eticheta CS-urile aparținând e-mulțimilor generate, s-a decis etichetarea, mai întâi, a CE-urilor aparținând synset-ului englezesc studiat. Aceste CE-uri vor fi etichetate în ordinea apariției lor cu numere întregi de la 1 la n (unde n reprezintă dimensiunea synset-ului, adică numărul de cuvinte pe care acesta îl conține). După etichetarea CE-urilor din synset-ul dat, CS-urile aparținând e-mulțimilor generate vor fi căutate în dicționarul bilingv corespunzător. De fiecare dată când un CE din synset-ul dat reprezintă, conform dicționarului, traducerea unui CS, acest CS primește eticheta CE-ului respectiv. Dacă cel puțin un cuvânt al unei e-mulțimi străine poate fi tradus prin unul dintre cuvintele synset-ului englezesc dat, atunci întreaga e-mulțime străină este inclusă în “lista candidaților”. Așa cum se remarcă în [Nikolov și Petrova, 01], atunci când devine completă, această listă a candidaților reprezintă cel mai important rezultat preliminar. Synset-ul străin adecvat va reprezenta o combinație a unor e-mulțimi aparținând acestei liste. Au fost concepute diverse *funcții de evaluare* care sortează

e-mulțimile generate, punându-le în evidență pe cele mai adecvate. Pentru a defini astfel de funcții de evaluare, ne vom referi, mai întâi, la următoarele concepte:

Definiția 2.2

Eticheta unei e-mulțimi reprezintă numărul de etichete atribuite cuvintelor aparținând acelei e-mulțimi.

Definiția 2.2

O e-mulțime este *neetichetată* dacă ea nu conține nici un cuvânt etichetat.

Orice cuvânt poate avea una sau mai multe etichete, precum și nici o etichetă. Cea mai simplă funcție de evaluare care este propusă în literatură de specialitate [Nikolov și Petrova, 01] are ca argument o e-mulțime și ca valoare însăși eticheta acelei e-mulțimi. O variantă a acestei funcții de evaluare este aceea care împarte numărul reprezentând eticheta e-mulțimii la cardinalul aceleiași e-mulțimi.

În ceea ce ne privește, am luat în considerație funcția de evaluare care este definită mai jos.

Fiecare CE aparținând synset-ului englezesc dat va avea o etichetă (reprezentată printr-un număr întreg de la 1 la n , unde n este dimensiunea synset-ului), iar etichetarea CS-urilor aparținând e-mulțimilor va fi efectuată în conformitate cu această etichetă. Etichetele CS-urilor care diferă de eticheta CE-ului corespunzător vor fi considerate ca reprezentând două puncte, în timp ce valoarea celorlalte etichete este de numai un punct. Valoarea funcției de evaluare relativ la o anumită e-mulțime este dată de numărul total de puncte corespunzător e-mulțimii împărțit la cardinalul acesteia.

Având definite toate conceptele necesare, putem, în cele ce urmează, să enunțăm algoritmul de generare a e-mulțimilor străine corespunzătoare unui synset englezesc dat:

Algoritmul 2.1

Input: Fișierul conținând synset-urile englezești și fișierele reprezentând cele două dicționare bilingve (spre exemplu, Dicționarul englez-francez și respectiv Dicționarul francez-englez).

1. Creează (prin consultarea dicționarului bilingv adecvat) e-mulțimile corespunzătoare fiecărui cuvânt al synset-ului englezesc dat.
2. Etichetează cuvintele englezești aparținând synset-ului englezesc dat.
3. Etichetează fiecare dintre e-mulțimile generate la pasul 1.
4. Îndepărtează toate mulțimile neetichetate.
5. Evaluează e-mulțimile (folosind etichetele atribuite și o funcție de evaluare).

Output: Lista sortată a e-mulțimilor corespunzătoare synset-ului englezesc dat.

Traducerile în limbă străină ale cuvintelor din synset-ul englezesc dat sunt extrase din dicționarul bilingv după cum urmează:

<i>cuvantel</i>	<i>sensl1; sensl2; ... ; senslm1</i>
.....
<i>cuvanten</i>	<i>sensn1; sensn2; ... ; sensnmn</i>

Mulțimea e-mulțimilor generată de Algoritmul 2.1 este de următoarea formă:

$$\{\{sens_{ij}\} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\}.$$

Synset-ul străin va fi generat folosind această mulțime.

În generarea automată a *synset-ului străin* corespunzător unui synset englezesc dat vom lua, de asemenea, în considerație

Observația 2.1

Dintre toate sensurile posibile ale unui cuvânt, numai unul se referă la un anumit concept (cărui îi corespunde un synset).

Folosind lista sortată a e-mulțimilor generată de Algoritmul 2.1 (cu alte cuvinte, e-mulțimile evaluate), sensul (mulțimea elementară) evaluat la cea mai mare valoare va fi ales corespunzător fiecărui cuvânt englezesc. Fie acest sens, corespunzător lui *cuvantej*, *sensji*.

Synset-ul străin va fi generat folosind e-mulțimile obținute prin intermediul Algoritmului 2.1, luând în considerație Observația 2.1 și conform

Algoritmul 2.2

Input: Lista sortată a e-mulțimilor generată de Algoritmul 2.1 corespunzător synset-ului englezesc dat [*cuvante1*, *cuvante2*, ..., *cuvanten*].

1. Determină synset-ul străin ca fiind de următoarea formă:

$$\{\text{sens1}_{i_1}\} \cup \{\text{sens2}_{i_2}\} \cup \dots \cup \{\text{sensn}_{i_n}\}, \quad 1 \leq i_j \leq m_j, \quad \forall j = \overline{1, n}$$

2. Șterge, din această reuniune, cuvintele aparținând mai multor e-mulțimi, astfel încât fiecare cuvânt să apară o singură dată.

Output: Synset-ul străin corespunzător synset-ului englezesc dat.

Algoritmii 2.1 și 2.2 au fost implementați în Prolog și testați de către noi, cu rezultate foarte bune, în cazul *substantivelor românești*. Toate testele efectuate au folosit WordNet 1.6 în format Prolog. Pentru a testa algoritmii prezentați aici s-au folosit fragmente de dicționare bilingve în format electronic. Au fost alese în mod aleator 200 de synset-uri de substantive englezești pentru care au fost generate automat synset-urile românești corespunzătoare. Întrucât majoritatea synset-urilor englezești conțin câte două cuvinte, eșantionul nostru de date a fost ales în funcție de același model. Astfel, din cele 200 de synset-uri englezești luate în considerație, 179 conțineau două substantive englezești, 4 conțineau câte trei substantive englezești, iar 17 synset-uri conțineau peste 3 substantive englezești (între 4 și 7 cuvinte). Numărul de e-mulțimi presupuse de experiment a fost de 616.

Datorită imperfecțiunii dicționarelor în format electronic existente, fragmentele de dicționare bilingve electronice folosite în cadrul experimentului au

fost create de noi, folosind [DRE, 73] și [DER, 74]. Propriile noastre dicționare bilingve în format electronic conțineau un total de 1164 cuvinte, dintre care 278 erau substantive englezești (corespunzând celor 200 de synset-uri englezești studiate), iar 886 de cuvinte reprezentau substantivele românești corespunzătoare. Fișierele conținând aceste dicționare bilingve reprezintă o parte a input-ului pentru programul ce implementează Algoritmul 2.1. Formatul în Prolog al acestor dicționare, utilizate în implementarea în Prolog a tuturor algoritmilor, poate fi văzut în §3.2.2. Synset-urile de substantive românești generate corespunzător celor 200 de synset-uri englezești studiate și care reprezintă output-ul Algoritmului 2.2 pot fi văzute pe web¹. Synset-urile românești generate au fost validate de către lingviști români, care au folosit cele mai complete dicționare bilingve, precum și glosa corespunzătoare, indicată în WordNet. Așa cum s-a mai menționat, această glosă conține explicația corespunzătoare unei serii sinonimice și, prin aceasta, conține meronimul sau conceptul “părinte”, aflat la un nivel superior în cadrul ierarhiei.

Atunci când algoritmul de traducere a fost testat referitor la substantivele românești, s-a constatat că, în anumite cazuri, Algoritmul 2.2 a generat mai mult de un singur synset românesc corespunzător celui englezesc dat. Aceasta a fost situația atunci când Algoritmul 2.1 a avut ca output o listă de e-mulțimi (corespunzătoare diferitelor sensuri ale aceluiași cuvânt) care fuseseră evaluate cu aceeași valoare. În acest caz, fiecare asemenea e-mulțime reprezenta un candidat și conducea la un synset românesc diferit. În astfel de cazuri, synset-ul românesc (sau, mai general, cel străin) corect va fi ales din lista de synset-uri generată de Algoritmul 2.2 în funcție de glosa synset-ului englezesc dat. Programul care implementează Algoritmul 2.2 trebuie, prin urmare, să furnizeze ca output și această glosă, necesară validării realizate de către lingviști.

În testarea algoritmilor relativ la substantivele românești s-a constatat că, pentru 86% dintre synset-urile englezești considerate, cele românești generate erau corecte. În celelalte cazuri, fie fuseseră generate mai multe synset-uri românești,

¹ <http://phobos.cs.unibuc.ro/roric/topic2.html>

printre care se afla și cel corect, fie synset-urile românești generate nu erau corecte, în special datorită informațiilor lipsă sau a celor greșite conținute în dicționarele bilingve. Considerăm acest rezultat foarte satisfăcător, întrucât este un fapt binecunoscut acela că nu se poate lucra 100% automat atunci când se operează cu resurse lingvistice.

De asemenea pentru a facilita experimentul, atunci când a fost ales eșantionul de synset-uri englezești, au fost înlăturate synset-urile conținând substantive proprii, cuvinte compuse și colocații. Acestea trebuie tratate separat și cu o contribuție mai semnificativă din partea lingviștilor. Totuși, considerăm că algoritmi prezentați sunt suficienți pentru a construi miezul synset-urilor corespunzătoare tuturor celor patru părți de vorbire, în mai mult sau mai puțin orice limbă diferită de limba engleză, cu condiția existenței dicționarelor bilingve complete în format electronic, corespunzător limbii studiate. Succesul metodei prezentate depinde în mod direct de existența acestor resurse în format electronic.

Așa cum se arată în [Nikolov și Petrova, 01], cel mai mare avantaj al algoritmului de traducere propus este abilitatea acestuia de a crea synset-uri străine care pot include cuvinte străine ce nu ar fi extrase din resursele constituind input-ul la primul pas al efectuării traducerii. Astfel, chiar dacă un cuvânt străin apare în Dicționarul englez-român, spre exemplu, dar lipsește din Dicționarul român-englez, există încă o mare șansă ca acest cuvânt să fie inclus în synset-ul rezultat. (Singura condiție necesară pentru aceasta este prezența în lista candidaților a unei e-mulțimi care include cuvântul respectiv). Acest fapt este extrem de important, mai ales având în vedere cât de incomplete sunt, de regulă, dicționarele bilingve. Algoritmul propus nu reprezintă, prin urmare, o simplă “traducere în oglindă”.

În mod evident, atunci când se utilizează Algoritmii 2.1 și 2.2 pentru diverse limbi particulare, diferite dificultăți vor interveni, în funcție de ceea ce este specific fiecărei limbi la nivel morfologic, semantic și derivativ. O dificultate generală, întâlnită indiferent de limba care este luată în considerație, constă în faptul că, într-o limbă dată, un anumit cuvânt se poate adesea referi la un concept extrem de general. El poate fi astfel pus în legătură cu mai multe cuvinte dintr-o

limbă diferită, limbă în care fiecare dintre cuvintele de care el este legat descrie un concept mult mai specializat. Aceasta este o chestiune extrem de importantă din punctul de vedere al oricărei abordări bazate pe WordNet, întrucât synset-urile din WordNet există în funcție de conceptele aflate la baza lor.

În ceea ce privește limba română¹, putem spune că principalele probleme care au apărut în traducerea automată a synset-urilor englezești și care pot genera situații în care programul nu lucrează corect, sunt reprezentate de așa-numiții falși prieteni, de colocații, de calcul lingvistic și de superioritatea polisemiei unor cuvinte englezești în raport cu corespondentele lor românești. Tot ca o concluzie vom observa faptul că cele mai multe probleme au apărut acolo unde synset-ul englezesc era compus dintr-un singur substantiv, cel mai adesea polisemantic, algoritmul neputând decide între sensuri. În viitor se impune, probabil, o tratare diferită a synset-urilor alcătuite dintr-un singur cuvânt².

În ciuda unor asemenea dificultăți, considerăm totuși algoritmi de traducere prezentați ca fiind adecvați pentru realizarea extragerii semiautomate a miezului unei WordNet străine pornind de la WordNet 1.6 pentru engleza americană. În cele ce urmează, vom stabili modul în care acest algoritm general trebuie îmbogățit pentru ca el să realizeze generarea semiautomată a **synset-urilor** și **cluster-elor de adjective** în alte limbi decât engleza.

3 Generarea synset-urilor și cluster-elor de adjective

3.1 Adjectivele în WordNet

WordNet împarte adjectivele în două mari clase³: adjective *descriptive* și *relaționale*. Adjectivele cromatice, care desemnează culori, sunt privite ca reprezentând un caz special.

¹ Pentru unele comentarii lingvistice referitoare la rezultatele obținute în cazul substantivelor românești, vezi <http://phobos.cs.unibuc.ro/roic/Ro/lingcom.html>.

² A se vedea tehnica de îmbogățire propusă în cazul synset-urilor de adjective (unice) în §3.2.1 al lucrării de față.

³ Prezentarea adjectivelor în WordNet este făcută aici conform [Miller et. al., 90].

Un *adjectiv descriptiv* este un adjectiv care atribuie unui substantiv o valoare a unui atribut. Cu alte cuvinte, x este *Adj* presupune existența unui atribut A astfel încât $A(x) = Adj$. De pildă, *scund* și *înalt* sunt valori ale atributului ÎNĂLȚIME. WordNet conține pointeri între adjectivele descriptive și synset-urile de substantive care se referă la atributele adecvate.

Organizarea semantică a adjectivei descriptive este fundamental diferită de cea a substantivelor. Relația hiponimică, care generează ierarhiile substantivale, nu este disponibilă și în cazul adjectivei. Organizarea semantică a adjectivei este gândită, în mod mai natural, ca reprezentând un hiperspațiu abstract cu N dimensiuni și nu un arbore ierarhic. Relația semantică de bază dintre adjectivele descriptive este *antonimia*.

Importanța antonimiei în organizarea adjectivei descriptive devine evidentă dacă avem în vedere faptul că funcția acestor adjective este aceea de a exprima valori ale atributelor, iar atributele sunt, în marea lor majoritate, bipolare. Adjectivele antonime exprimă valori opuse ale unui atribut. Spre exemplu, antonimul lui *greu* este *ușor*, care exprimă o valoare de la polul opus al atributului GREUTATE. În WordNet această opoziție binară este reprezentată prin intermediul pointerilor reciproci etichetați: *greu!->ușor* și *ușor!->greu*. În implementarea în Prolog a WordNet, pe care am folosit-o pentru acest studiu, operatorul **ant** specifică cuvinte antonime și toate faptele Prolog care utilizează acest operator sunt incluse în fișierul **wn_ant.pl**. Adjectivele descriptive care nu posedă antonime directe sunt privite ca având antonime indirecte, datorită similarității lor semantice cu adjective care au antonime directe. Un pointer de similaritate a fost folosit pentru a indica faptul că toate adjectivele care nu posedă antonime sunt similare ca sens cu adjective care au antonime. În implementarea în Prolog a WordNet operatorul **sim** specifică faptul că două synset-uri sunt similare ca înțeles și toate faptele Prolog care utilizează acest operator sunt incluse în fișierul **wn_sim.pl**.

Prin urmare, adjectivele descriptive atribuie “substantivelor cap” corespunzătoare valori ale unor atribute cel mai adesea bipolare și, în consecință,

sunt organizate în termeni de *opoziție binară (antonimie)* și *similaritate a sensului (sinonimie)*.

Gross, Fischer și Miller (1989) propun ca synset-urile de adjective să fie privite ca niște *cluster-e de adjective* asociate prin similaritate semantică unui adjectiv focal, care face legătura dintre cluster-ul respectiv și un cluster contrastant, aflat la polul opus al atributului. Tot Gross, Fischer și Miller fac distincția între antonimele directe, cum ar fi *heavy / light (greu / ușor)*, care sunt opuse conceptual, reprezentând, în același timp, perechi lexicale și între antonimele indirecte, cum ar fi *heavy / weightless (greu / lipsit de greutate)*, care sunt opuse conceptual fără a reprezenta perechi lexicale. Sub această formulare, toate adjectivele descriptive au antonime; cele cărora le lipsesc antonimele directe au antonime indirecte i.e. sunt sinonime ale unor adjective care au antonime directe.

În WordNet antonimele directe sunt reprezentate prin pointerul de antonimie '!->'; antonimele indirecte sunt moștenite prin similaritate, care este indicată prin pointerul de similaritate '&->'. Configurația rezultată este ilustrată în Figura 1, pentru cluster-ul de adjective centrat în jurul antonimelor directe *wet / dry (umed / uscat)*, care definesc atributul WETNESS sau MOISTNESS (UMEZEALĂ), un exemplu oferit în [Miller et. al., 90] și folosit de numeroși autori. Atunci când se analizează acest cluster de adjective englezești se poate observa, spre exemplu, că *moist* nu are un antonim direct, dar antonimul său indirect poate fi găsit via drumul *moist&->wet!->dry*.

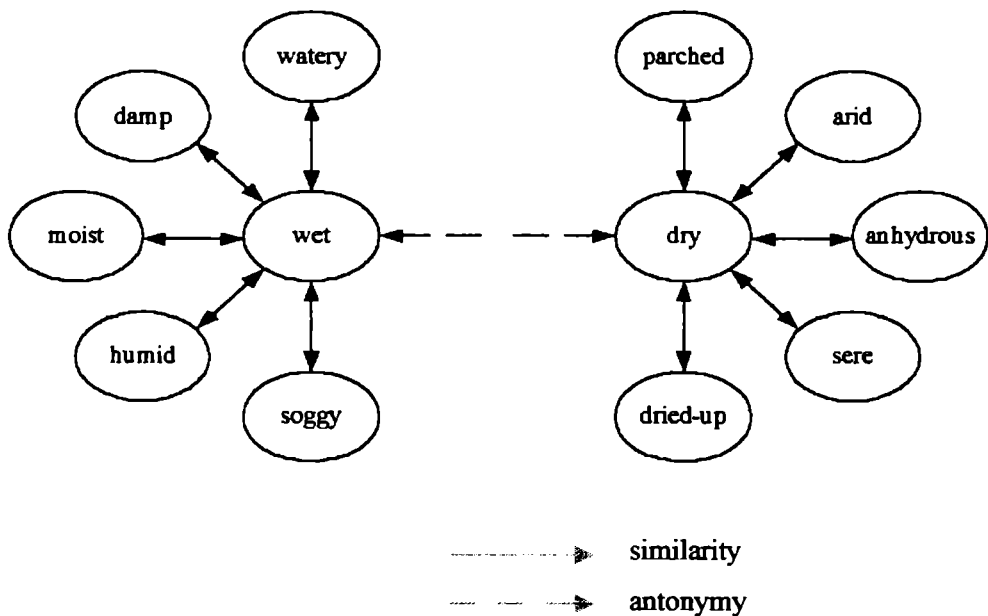


Figura 1. Structură bipolară de adjective

Corespunzător Figurii 1, care prezintă, în mod intuitiv, structura unui cluster de adjective în WordNet, se obține următorul cluster bipolar, având capul reprezentat de perechea de antonime *wet / dry*:

```
[{ [WET, DRY, !] watery,& damp,& moist,& humid,& soggy,& }
{ watery, tearful, teary, wet, & }
{ damp, wet,& }
{ moist, wet,& }
{ humid, muggy, steamy, sticky, sultry, wet,& }
{ soggy, saturated, sodden, waterlogged, wet,& }
-
{ [DRY, WET,!] parched,& arid,& anhydrous,& sere,& dried-up,& }
{ parched, dehydrated, desiccated, dry,& }
{ arid, waterless, dry,& }
{ anhydrous, dry,& ((chem) with all water removed)}
{ sere, shriveled, withered, wizened, dry,& (used of vegetation)}
{ dried-up, dry,& ("a dry water hole") }].
```

Se poate observa că un cluster are două părți distincte. Fiecare jumătate a unui cluster începe cu un synset numit *synset cap*. Primele două elemente ale synset-ului cap reprezintă perechea de antonime care definește cluster-ul și sunt scrise cu majuscule. Această pereche de antonime este urmată de adjective reprezentând pointeri de similaritate (*watery,& damp,& etc.*), unul către fiecare synset similar ca sens din jumătatea respectivă a cluster-ului. Fiecare astfel de synset conține un pointer reciproc pentru întoarcerea la synset-ul cap. Se observă, de asemenea, că pointerii de similaritate care intervin în synset-ul cap sunt, de fapt, cuvinte care ocupă prima poziție în cadrul synset-urilor înrudite prin similaritate cu synset-ul căruia îi aparține adjectivul ce are un antonim direct. Cele două părți distincte ale cluster-ului sunt separate printr-o cratimă, iar întreaga structură este inclusă între paranteze drepte. Fiecare cluster bipolar este de sine stătător, iar codificarea este restricționată la relațiile din interiorul cluster-ului.

Semnificația excepțiilor care, fără îndoială, există nu este majoră și considerăm, împreună cu autorii WordNet [Miller et. al., 90], că modelul prezentat acoperă marea majoritate a adjectivelor descriptive englezești. Importanța relației de similaritate este evidentă.

În WordNet relațiile semantice sunt reprezentate printr-o pereche de id-uri de synset (*synset_id*-uri), pereche în cadrul căreia primul *synset_id* reprezintă, în general, sursa relației, iar cel de-al doilea ținta acesteia.

Un *synset_id* este un câmp de 9 bytes în care primul byte definește categoria sintactică a synset-ului, iar cei 8 bytes rămași reprezintă un *synset_offset*. În versiunea Prolog a bazei de date WordNet, *synset_id*-urile sunt folosite ca identificatori unici de synset-uri. Așa cum s-a remarcat deja, în implementarea în Prolog a WordNet operatorul **sim** este folosit pentru a desemna relația de similaritate, ca în exemplul **sim(302425348, 302425924)**. În general,

sim(synset_id, synset_id).

este o faptă Prolog care specifică faptul că al doilea synset este similar ca sens cu primul. Aceasta înseamnă că cel de-al doilea synset este un satelit al primului

synset, care reprezintă capul cluster-ului. Această relație este valabilă numai pentru synset-uri de adjective conținute în cluster-e de adjective.

În WordNet *adjectivele relaționale*, care au fost discutate, mai întâi, pe larg, de către Levi (1978), înseamnă ceva de tipul “relativ la, referitor la, ținând de sau asociat cu” un substantiv. Adjectivele relaționale se deosebesc de cele descriptive prin aceea că ele nu pot fi puse în legătură cu un atribut. Prin urmare, ele nu se referă la o proprietate a substantivului cap. Întrucât adjectivele relaționale nu au antonime, ele nu pot fi încorporate în cluster-ele care caracterizează adjectivele descriptive. WordNet menține un fișier separat de adjective relaționale cu pointeri către substantivele corespunzătoare. Fiecare synset constă din unul sau mai multe adjective relaționale urmate de un pointer către substantivul corespunzător.

În cele ce urmează, ne vom ocupa de generarea semiautomată a *cluster-elor de adjective* din alte limbi decât engleza și, prin urmare, ne vom referi exclusiv la adjectivele descriptive, care pot fi organizate sub forma unei structuri de acest tip.

3.2 Generarea semiautomată a synset-urilor de adjective

3.2.1 Algoritmul

Pentru a traduce într-o limbă străină synset-urile de adjective englezești, au fost folosiți Algoritmii 2.1 și 2.2. Atunci când se traduce din engleză în orice altă limbă, *id-ul* asociat unui synset nu se modifică. Aceasta înseamnă că relația de similaritate existentă între doua synset-uri englezești se va menține după efectuarea traducerii și va interveni, în egală măsură, între synset-urile de adjective din limba străină respectivă.

O problemă specială o ridică synset-urile conținând un singur cuvânt. În acest caz este imposibil de știut care dintre sensurile cuvântului (presupunând că acesta este polisemantic, cum se întâmplă cel mai adesea în limba engleză) intervine în crearea synset-ului studiat. Sensul în cauză poate fi dedus numai pe

baza glosei. Pentru astfel de cazuri, propunem folosirea unei strategii constând în îmbogățirea synset-ului dat cu noi adjective, care vor sugera sensul unicului adjectiv ce intervine în synset-ul inițial. Noile adjective, folosite pentru îmbogățire, sunt obținute prin intermediul relației de similaritate, care există în WordNet cu specială referire la synset-urile de adjective. Astfel, pentru a îmbogăți synset-ul dat cu noi cuvinte, vor fi alese adjectivele care intervin pe prima poziție în cadrul synset-urilor legate de cel inițial prin intermediul relației de similaritate. Aceste cuvinte vor fi adăugate synset-ului inițial începând cu cea de-a doua poziție. Această idee a fost inspirată de modul în care sunt organizate și structurate în WordNet cluster-ele de adjective.

Lista e-mulțimilor legate de synset-ul englezesc inițial va fi generată folosind Algoritmul 2.1. Pentru crearea synset-ului de adjective străin reprezentând traducerea celui englezesc dat, se va folosi Algoritmul 2.2 (care efectuează un proces de backtracking). Acesta va combina e-mulțimile corespunzând fiecăruia dintre cuvintele ce intervin în synset-ul englezesc care au fost evaluate maximal. În acele cazuri în care mai multe e-mulțimi corespunzătoare aceluiași cuvânt englezesc au fost evaluate maximal, Algoritmul 2.2 va genera mai multe synset-uri de adjective străine. Decizia finală cu privire la cel corect se va face cu luarea în considerație a glosei.

Pentru a ilustra modul în care lucrează Algoritmii 2.1 și 2.2 în cazul synset-urilor de adjective, vom lua în considerație synset-ul englezesc având *id*-ul 302428719 și conținând unicul adjectiv *sticky*. Vom realiza traducerea în limba română a acestui synset. Să observăm, mai întâi, că limba țintă aleasă nu este esențială pentru ceea ce dorim să demonstrăm aici. Rezultatele care vor fi comentate reprezintă output-ul diverselor programe Prolog care implementează algoritmii amintiți.

Întrucât synset-ul englezesc dat conține un singur cuvânt, el va fi îmbogățit, așa cum s-a menționat, în conformitate cu relația de similaritate. După căutarea în baza de date se ajunge la concluzia că unica relație de similaritate (desemnată de operatorul **sim**) este următoarea:

precum și simetrica ei. Synset-ul având *id* = 302425348 conține unicul adjectiv *wet*. Synset-ul englezesc dat este, prin urmare, îmbogățit cu acest adjectiv. E-mulțimile evaluate obținute corespunzător synset-ului îmbogățit, atunci când se folosește pentru Algoritmul 2.1 funcția de evaluare menționată în §2, sunt următoarele:

evset (302428719, sticky, 1.0, [lipicios, cleios, vascos]).

evset (302428719, sticky, 1.0, [umed, cetos]).

evset (302428719, wet, 1.0, [umed, jilav, ud]).

evset (302428719, wet, 0.6666666666666666, [ploios, umed, igrasios]).

Aici *evset* este un operator care desemnează e-mulțimi evaluate. Primul câmp reprezintă *id*-ul synset-ului, cel de-al doilea este textul ASCII corespunzător cuvântului, așa cum este el introdus de către lexicograf, cel de-al treilea furnizează valoarea funcției de evaluare, iar ultimul desemnează mulțimea evaluată de cuvinte străine.

În acest caz programul care implementează Algoritmul 2.2 are următorul output:

Synset englezesc: [sticky]

Glosa:

(moist as with undried perspiration and with clothing sticking to the body; “felt sticky and chilly at the same time”)

Romanian synset: [[lipicios,cleios,vascos,umed,jilav,ud], [umed,cetos,jilav,ud]]

Se observă că au fost generate două synset-uri românești posibile. Doar unul dintre ele corespunde însă sensului lui *sticky* care se referă la conceptul aflat la baza synset-ului având *id* = 302428719. Synset-ul străin (în acest caz românesc) corect poate fi ales cu ușurință în conformitate cu glosa corespunzătoare.

O astfel de îmbogățire cu cuvinte suplimentare, provenite din synset-uri înrudite prin relația de similaritate cu cel inițial, nu este întotdeauna necesară. Atunci când ea este totuși efectuată, posibilitatea obținerii unor synset-uri străine vide (datorată generării în exclusivitate a e-mulțimilor neetichetate) este considerabil redusă. Operația de îmbogățire poate produce o ușoară deplasare a sensului, în ceea ce privește conceptul aflat la baza synset-ului englezesc inițial. Cum însă această relație, tipică pentru adjectivele descriptive, se referă numai la concepte similare, recomandăm utilizarea strategiei descrise. Se pot realiza atât traducerea cu îmbogățire, cât și cea fără îmbogățire, relativ la un același synset, oferind lingviștilor posibilitatea de a compara rezultatele (synset-urile străine) obținute și eliminând necesitatea consultării glosei (care poate fi totuși folosită ca instrument suplimentar de lucru).

3.2.2 Implementare Prolog

Baza de date WordNet în format Prolog este conținută în fișierele **wn_*.pl** și folosește *synset_id* - uri ca identificatori unici de synset-uri. A fost creat câte un fișier separat pentru fiecare relație din WordNet, ceea ce îi oferă utilizatorului posibilitatea de a încărca numai acele părți ale acestei imense baze de date care îi sunt necesare la un moment dat. Fiecare fișier al bazei de date Prolog conține informații corespunzătoare synset-urilor și sensurilor cuvintelor incluse în baza de date WordNet. Fiecare linie a unui fișier conține un operator care va corespunde unei relații din WordNet. Toate liniile cu aceeași valoare a operatorului sunt incluse în fișierul **wn_operator.pl**. Forma generală a unei linii dintr-un fișier al bazei de date Prolog este

operator(camp1,...,campn).

Ea nu conține nici un spațiu și se termină cu un caracter de tip “newline”.

Un operator **s** este prezent pentru fiecare sens al unui cuvânt din WordNet, în timp ce operatorul **g** specifică glosa corespunzătoare unui synset. Fișierul

wn_s.pl conține toate synset-urile din WordNet, în timp ce fișierul **wn_g.pl** conține toate glosele.

Fișierul **wn_s.pl** ce conține toate synset-urile din WordNet este format din fapte Prolog având forma generală

s(synset_id, nr_cuv, 'cuvant', tip_ss, nr_sens, stare_eti).

unde:

- *synset_id* este identificatorul synset-ului;
- *nr_cuv* este numărul cuvântului în cadrul synset-ului (comunică al câatlea cuvânt este 'cuvânt' în cadrul synset-ului și se preia din matricea lexicală, din cadrul *liniei*);
- '*cuvant*' este textul ASCII al cuvântului la care se referă fapta Prolog;
- *tip_ss* este un cod de un caracter ce indică tipul synset-ului, cod care specifică partea de vorbire a lui 'cuvânt' (spre exemplu, **n** pentru substantiv, de la engl. "noun");
- *nr_sens* este un număr natural care desemnează numărul sensului (preluat din cadrul *coloanei* matricii lexicale);
- *stare_eti* poate fi 1 sau 0, coduri având următoarea semnificație:
 - 1 - sensurile cuvântului sunt ordonate după frecvența utilizării;
 - 0 - sensurile nu sunt ordonate după frecvența utilizării, ci conform altor criterii.

Un operator de tip **s** este prezent pentru fiecare sens din WordNet.

Un exemplu de o asemenea faptă Prolog, corespunzătoare adjectivului englezesc *abridged*, este următoarea:

s(300004481,1,'abridged',a,1,0).

Datorită dimensiunilor bazei de date WordNet, pentru a micșora timpul calculator, s-a realizat "spargerea" fișierului **wn_s.pl** în patru fișiere mai mici, fiecare dintre acestea conținând toate faptele Prolog care se referă la o aceeași parte

de vorbire. Unul dintre aceste fișiere constituie întotdeauna o parte a input-ului tuturor programelor care realizează generarea automată a synset-urilor străine.

Pentru a micșora și mai mult timpul calculator, după eliminarea informațiilor care nu erau necesare scopului propus, fișierul **wn_s.pl** conținând numai fapte Prolog referitoare la adjective a fost prelucrat în continuare. Au fost scrise programe de calculator necesare obținerii unei forme simplificate a faptelor Prolog referitoare la adjective.

Mai întâi au fost eliminați parametrii *a* și/sau *s*, care desemnează partea de vorbire. Cel de-al doilea parametru, precum și ultimii doi au fost, de asemenea, considerați ca nefiind utili scopului propus. Au fost eliminate majusculele și simbolurile speciale cum ar fi apostroful, precum și faptele Prolog care conțineau substantive proprii și colocații. S-a considerat că acestea din urmă, extrem de importante, ar trebui să facă obiectul unui studiu separat. Fișierul rezultat, care conține synset-uri de adjective, include fapte Prolog de forma următoare:

s(300003469,'emergent').

s(300003469,'emerging').

s(300003469,'nascent').

În fine, fișierul rezultat a fost procesat astfel încât să se obțină o unică faptă Prolog corespunzătoare fiecărui synset. În cazul exemplului de față, a fost creată următoarea faptă Prolog:

s(300003469, [emergent, emerging, nascent]). (1)

Dorim să remarcăm în mod special faptul că, atunci când se adaugă cuvintele pentru crearea synset-ului, este esențială menținerea ordinii în care acestea intervin, datorită deciziei noastre de a alege întotdeauna adjectivul plasat pe prima poziție atunci când se realizează îmbogățirea synset-urilor conținând un singur adjectiv, prin intermediul relației de similaritate. Modul în care sunt construite cluster-ele de adjective face, de asemenea, necesar acest lucru. Fișierul conținând fapte Prolog de forma (1), câte o astfel de faptă Prolog corespunzător fiecărui synset de adjective

din WordNet, constituie o parte a input-ului tuturor programelor care implementează atât Algoritmul 2.1, cât și Algoritmul 2.2.

Ne vom referi la acest fișier ca la “*fișierul de synset-uri curățat și combinat*”. Astfel de fișiere curățate și combinate au fost folosite atât în cazul substantivelor, cât și în cel al adjectivelor, atunci când au fost testate programele care implementează algoritmi propuși. Comentariile de față se referă, cu precădere, la adjective, întrucât procesarea synset-urilor de adjective a necesitat cațiva pași suplimentari, după obținerea acestui tip de fișier.

Fișierul curățat și combinat este o parte a input-ului programelor care implementează Algoritmul 2.1 și Algoritmul 2.2 atunci când se efectuează o traducere fără îmbogățire.

O altă parte importantă a input-ului tuturor programelor o constituie cele două dicționare bilingve. În cazul exemplului la care ne-am referit în §3.2.1 (cuvântul *sticky*), o intrare în Dicționarul englez-român este de forma

sticky lipicios , cleios , vascos , ; umed , cetos

Au fost utilizați numai doi separatori, “virgulă” și “punct și virgulă”. În timp ce virgula separă sinonime care se referă la același sens al unui cuvânt, simbolul “punct și virgulă” este folosit ca separator între sensuri. Fiecare separator este precedat și urmat de un blank. Cuvântul englezesc este singurul urmat de mai multe blankuri. Următoarea faptă Prolog a fost obținută corespunzător unei asemenea intrări în dicționar:

word1(sticky, [[lipicios, cleios, vascos], [umed, cetos]]).

Predicatul de mai sus (*word1*) are doi parametri, și anume cuvântul englezesc și lista e-mulțimilor corespunzătoare lui. Această listă este inclusă între paranteze drepte. Fiecare e-mulțime este inclusă, la rîndul ei, între paranteze drepte. Dicționarul român-englez este alcătuit din fapte Prolog similare.

În pagina web a proiectului¹ veți găsi **implementarea în Java** a tuturor algoritmilor propuși, care utilizează un format similar al dicționarelor bilingve, după cum este descris la această adresă. Programul corespunzător, GenSynsets, folosește rezultatele obținute în cadrul proiectului JWordNet. JWordNet reprezintă o interfață de sine stătătoare, orientată obiect, scrisă în Java, ce implementează diversele entități lexicale și semantice din WordNet. Amănunte despre programul GenSynsets pot fi aflate în pagina web a proiectului.

Următorul pas în pregătirea implementării Algoritmului 2.1 constă în îmbogățirea automată a acelor synset-uri de adjective ce conțin un singur cuvânt, prin intermediul relației de similaritate. Programul care realizează această operație primește ca input fișierul de synset-uri curățat și combinat, precum și fișierul **wn_sim.pl** al bazei de date Prolog, care conține faptele Prolog corespunzătoare relațiilor de similaritate existente. Output-ul este reprezentat de un fișier care conține fapte Prolog de următoarea formă:

$$stl(synset_id, synset).$$

Acest fișier include toate synset-urile de adjective din WordNet, cele care conțineau un singur cuvânt fiind acum îmbogățite (în conformitate cu similaritățile existente). Acesta este, de fapt, **fișierul cu synset-uri final**, care reprezintă input-ul programelor de calculator ce implementează Algoritmul 2.1 și respectiv Algoritmul 2.2, atunci când se efectuează o traducere cu îmbogățire.

În ceea ce privește implementarea în Prolog a relațiilor semantice și lexicale din WordNet, această problemă este discutată în §3.1 și §3.3, atunci când se face referire la operatorii **sim** și respectiv **ant**.

3.3 Generarea semiautomată a cluster-elor de adjective

Traducerea cluster-elor de adjective englezești este complet asigurată de către traducerea synset-urilor englezești de adjective și a relației **ant** (care

¹ <http://phobos.cs.unibuc.ro/roric/topic2.html>

desemnează antonime). Întrucât traducerea synset-urilor de adjective a fost deja discutată în §3.2, ne vom referi, în cele ce urmează, la traducerea relației **ant**. Aceasta este o problemă foarte importantă, având în vedere faptul că, pentru un mare număr de limbi, dicționarele de antonime în format electronic sunt inexistente.

În versiunea Prolog a bazei de date WordNet, care s-a folosit aici, relațiile semantice sunt reprezentate printr-o pereche de *synset_id*-uri, în care primul *id* este, în general, sursa relației, iar cel de-al doilea reprezintă ținta, ca în cazul operatorului deja menționat **sim**. Dacă însă sunt prezente două perechi *synset_id*, *w_num*, atunci operatorul reprezintă o relație lexicală între forme lexicale, în care *w_num* specifică numărul cuvântului, în cazul unui anumit cuvânt dintr-un anumit synset. Dacă este prezent, *w_num* indică la care dintre cuvintele synset-ului se face referire. Operatorul **ant**, de pildă, specifică cuvinte antonime în următoarea formă:

ant(synset_id,w_num,synset_id,w_num).

Astfel, semnificația faptei Prolog

ant(302425348, 1, 302429323, 1).

este aceea că primul cuvânt al synset-ului având *id*-ul 302425348 și primul cuvânt al synset-ului având *id*-ul 302429323 sunt *antonime directe*. Aceasta este o relație lexicală care este valabilă pentru aproape toate părțile de vorbire, dar care este esențială în formarea cluster-elor de adjective. În cazul fiecărei perechi de antonime sunt date ambele relații (i.e. fiecare pereche *synset_id,w_num* desemnează atât un cuvânt sursă, cât și un cuvânt țintă).

Atunci când este studiat conținutul fișierului **wn_ant.pl** al bazei de date WordNet în format Prolog, fișier care conține toate faptele Prolog referitoare la cuvinte antonime, se observă cu ușurință că marea majoritate a acestor fapte stabilesc relații de antonimie directă între cuvinte aflate pe prima poziție în cadrul synset-urilor cărora le aparțin. Există mai puțin de 15 excepții de la această regulă. Aceste excepții pot fi cu ușurință procesate de către un operator uman care poate

reține noile poziții ale adjectivelor având antonime directe. În aceste condiții, considerăm că putem formula

Observația 3.1

Primul cuvânt al unui synset englezesc de adjective este cel care poate avea un antonim direct.

Să presupunem acum că toate synset-urile de adjective străine (traduse) există, corespunzător unei limbi date și că ele aparțin fișierului numit **wn_strans.pl**. În urma generării acestui fișier, folosind Observația 3.1 și aplicând algoritmul de traducere, putem formula algoritmul de generare a cluster-elor de adjective străine corespunzătoare celor englezești:

Algoritmul 3.1

Input: Fișierele **wn_ant.pl**, **wn_sim.pl** și **wn_strans.pl**

Pentru fiecare *pereche de synset-uri* desemnată de către fiecare faptă Prolog a fișierului **wn_ant.pl**, execută pașii 1.-5.:

1. Caută în fișierul **wn_strans.pl** și găsește synset-urile străine reprezentând traduceriile celor englezești date.
2. Corespunzător fiecărui synset străin găsit în fișierul **wn_strans.pl** la pasul 1., reține primul cuvânt al aceluia synset. (Această pereche de cuvinte va constitui capul cluster-ului străin obținut prin traducere).
3. Corespunzător aceleiași perechi de cuvinte, caută în fișierul **wn_sim.pl** și ia în considerație clauzele **sim** corespunzătoare fiecăruia dintre cele două synset-uri cărora le aparțin cele două cuvinte ale capului de cluster.
4. Ia în considerație toate synset-urile desemnate de clauzele **sim** alese la pasul 3., synset-uri având cel de-al doilea *id* care intervine în clauză.

Gasește synset-urile străine reprezentând traduceri ale acestora în fișierul **wn_strans.pl**.

5. Adaugă fiecare prim cuvânt al acestor synset-uri străine în capul de cluster, împreună cu pointerul &.
6. Adaugă fiecare synset străin “similar”, plasând la sfârșitul acestuia pointerul reciproc de similaritate.

Output: Un fișier conținând toate cluster-ele de adjective străine.

Algoritmul 3.1 va genera cluster-e străine de adjective cu o structură bipolară ca aceea descrisă în §3.1 și ilustrată în Figura 1. Corespunzător acelui cluster, cu alte cuvinte celui care are ca synset cap perechea de antonime [WET, DRY, !] și respectiv [DRY, WET, !], a fost generat următorul cluster românesc:

```
{ [UMED, USCAT, !] inourat,& stropit,& vascos,& umed,& umed,&
cetos,& lipicios,& ploios,& lipicios,& umed,& }
{ inourat, stropit, umezit, umed,& }
{ stropit, smaltat, umed,& }
{ vascos, cleios, lipicios, umed,& }
{ umed, igrasios, jilav, ud, umed,& }
{ umed, jilav, ud, umed,& }
{ cetos, aburit, umed, jilav, ud, umed,& }
{ lipicios, cleios, vascos, umed,& }
{ ploios, umed,& }
{ lipicios, cleios, vascos, umed, jilav, ud, umed,& }
{ umed, jilav, ud, umed,& }
-
{ [USCAT, UMED, !] arid,& uscat,& secat,& uscat,& uscat,& uscat,& }
{ arid, uscat, sec, uscat,& }
{ uscat, arid, sterp, sec, uscat,& }
```

```
{ secat, uscat,& }  
{ uscat, ofilit, vestejit, zbarcit, uscat,& }  
{ uscat, arid, sterp, sec, uscat,& }  
{ uscat, arid, insetat, uscat,& }]
```

Menționăm că, în cazul de mai sus, nu au fost folosite toate relațiile de similaritate existente, întrucât scopul nostru nu a fost decât acela de a oferi un exemplu referitor la *tipul* de structură pe care o generează Algoritmul 3.1.

Cluster-ul românesc pe care l-am obținut nu ilustrează decât procedeele de codificare de bază folosite în WordNet-ul american, care au fost preluate de către noi. În acest stadiu de început al studiului nostru, am fost preocupați numai de crearea structurii de cluster *de tip WordNet* și nu am încercat să facem distincția între diferitele sensuri secundare sau privilegii de apariție. Nu am încercat, de asemenea, să indicăm limitările anumitor adjective relativ la pozițiile sintactice pe care acestea pot să le ocupe, un tip de limitare care în WordNet este codificat relativ la adjective individuale. Aceste aspecte pot fi tratate cu relativă ușurință, după stabilirea algoritmului de bază. Ele, ca și altele, vor face obiectul unor studii viitoare.

Diverse dificultăți de natură lingvistică vor fi întâmpinate, în mod evident, în funcție de limba țintă aleasă. Synset-uri străine identice ar putea fi generate, spre exemplu, de către Algoritmul 3.1, corespunzător unor synset-uri englezești diferite și, prin urmare, corespunzător unor sensuri și concepte diferite. Aceasta este situația atunci când un adjectiv polisemantic englezesc va avea în engleză unul sau mai multe sensuri care în limba țintă nu există, fenomen lingvistic care este numit *împrumut semantic* și care este un aspect al calcului în general. Validarea lingvistică a output-ului programelor care implementează Algoritmul 3.1, sau orice alt algoritm de aceeași natură, va fi, prin urmare, întotdeauna necesară. Cu toate acestea, considerăm că Algoritmul 3.1 acoperă marea majoritate a cazurilor existente, atunci când se lucrează cu cluster-e de adjective de tip WordNet.

4 Considerații finale

WordNet a fost recunoscută ca o resursă extrem de valoroasă de către comunitățile științifice ale tehnologiei limbajului și procesării cunoștințelor. Majoritatea cercetătorilor care utilizează WordNet în special în domeniul inteligenței artificiale privesc această rețea în primul rând ca pe o *bază de cunoștințe* și o folosesc în consecință. Procesarea cunoștințelor a dobândit noi dimensiuni în S.U.A. datorită existenței WordNet. Utilitatea ei în diferite aplicații practice a fost citată în peste 200 de articole științifice și au fost implementate sisteme care o folosesc. Multe grupuri de cercetători și-au exprimat interesul în aplicațiile bazate pe WordNet din diferite domenii, cum ar fi: regăsirea informației, extragerea informației, dezambiguizarea sensului cuvintelor, generarea limbajului natural, învățarea, achiziția de cunoștințe etc.

Comunitatea științifică din domeniul limbajului natural a încurajat și încurajează dezvoltarea unor rețele de tip WordNet pentru alte limbi decât engleza, concentrându-se, în același timp, asupra posibilității generării automate a unor asemenea baze de date de mari dimensiuni. Principalul motiv al acestui efort îl constituie dorința și necesitatea de a crea o *infrastructură ontologică uniformă* relativ la cât mai multe limbi. Aceasta va simplifica traducerea dintr-o limbă în alta și va facilita utilizarea acelorași scheme de raționament, precum și a algoritmilor concepuți în legătură cu rețeaua americană WordNet.

Precizări

1. Autoarea dorește să mulțumească Prof. dr. Theodor Hristea de la Facultatea de Litere a Universității din București pentru a fi oferit o constantă și prețioasă consultanță lingvistică referitoare la WordNet, precum și pentru validarea synset-urilor și cluster-elor românești generate de programele care implementează algoritmi descriși aici.

2. Mulțumirile sale sunt adresate, de asemenea, studenților de la programul de studii aprofundate Cristina Vata, Claudia Burtea și Mihai Sima de la Facultatea de Matematică a Universității din București, precum și studentei Raluca Elena Galatanu de la Facultatea de Litere a aceleiași universități, pentru a fi asistat, cu mult devotament, echipa RORIC-LING, pe întreg parcursul celei de-a doua părți a proiectului BALRIC-LING.

Bibliografie

[Fellbaum, 98] Fellbaum,C. (Ed.): “WordNet: An Electronic Lexical Database”; The MIT Press, Cambridge/London/England (1998).

[Harabagiu, 99] Harabagiu, S.: “Lexical Acquisition for a Romanian WordNet”; Proc. EUROLAN '99, Iași, România (1999).

[Miller et. al., 90] Miller,G.A., Beckwith,R., Fellbaum,C., Gross,D., Miller,K.J.: “Introduction to WordNet: an on-line lexical database”; International Journal of Lexicography, 3,4 (1990), 235-244.

[Nikolov and Petrova, 00] Nikolov, T., Petrova, K.: “Building and Evaluating a Core of Bulgarian WordNet for Nouns”; OntoLex '2000 Report, Sozopol, Bulgaria (2000).

[Nikolov and Petrova, 01] Nikolov, T., Petrova,K.: “Towards Building Bulgarian WordNet”; Proc. RANLP'01, INCOMA Ltd., Tzigov Chark, Bulgaria (2001), 199-203.

[DRE, 73] Levițchi,L., “Dicționar român-englez” (ed.a III-a); Editura Științifică, București (1973).

[DER, 74] “Dicționar englez-român”; Editura Academiei R.S.R., București (1974).

GenSynsets – implementarea Java a algoritmilor

George Ungureanu

GenSynsets este un instrument conceput pentru a facilita dezvoltarea de *WordNet-uri* si pentru alte limbi in afara de limba engleza.. Acesta implementeaza algoritmii descrisi in articolul "Asupra generarii semiautomate a synset-urilor si cluster-elor de tip WordNet cu speciala referire la limba romana". **GenSynsets** poate fi folosit pentru orice limba pentru care exista dictionare bilingve in format electronic. Programul a fost testat pentru limba romana, iar rezultatul corespunzator (un fisier XML) poate fi consultat pe Web.

Instalare

GenSynsets este scris in limbajul de programare Java si ruleaza pe platforme Java 2.

Cerinte:

1. Pentru a putea rula, **GenSynsets** are nevoie de mediul de executie Java 2. Deci sistemul de operare al calculatorului pe care se doreste instalarea trebuie sa fie unul pentru care exista o implementare a lui Java 2 (Windows 95/98/ME/NT/000, majoritatea versiunilor de Unix, MacOS X).
2. Sistemul pe care se va instala **GenSynsets** trebuie sa fie unul suficient de puternic (viteza procesor, memorie). In cazul unui PC sunt necesare minimum 133 MHz, 32M RAM.

Instalare:

1. Instalati Java 2 pe sistemul dumneavoastra. Daca Java 2 este deja instalat, sariti peste acest pas. Pentru platformele Windows, Linux, Solaris, kiturile de instalare corespunzatoare se pot gasi la java.sun.com. Se poate instala intreg mediul de dezvoltare JDK sau doar mediul de executie JRE. Se recomanda folosirea versiunii 1.3 sau a uneia mai noi.
2. Asigurati-va ca este pusa in PATH calea catre executabilul java. In Windows 95/98/NT aceasta se poate face punand in **autoexec.bat** o linie de forma:

```
SET PATH=c:\calei; %PATH%
```

unde `c:\cale` se inlocuieste cu calea actuala catre directorul unde se afla executabilul `java`. Dupa aceasta operatie (si restartarea sistemului), ar trebui ca la comanda (indiferent de directorul din care este data aceasta):

```
java -version
```

raspunsul sistemului sa fie asemanator cu:

```
C:\>java -version
java version "1.3.0"
Java(TM) 2 Runtime
Environment, Standard Edition
(build 1.3.0-C)
Java HotSpot(TM) Client VM
(build 1.3.0-C, mixed mode)
```

```
C:\>
```

3. Desfaceti arhiva `gensynsets.zip` si plasati continutul acesteia oriunde doriti in structura de directoare.

Nota: Instructiunile de mai sus s-au concentrat mai mult pe instalarea sub sistemul de operare Windows. Pentru a instala si rula **GenSynsets** pe orice sistem, in esenta, trebuie sa instalati mediul Java 2 pe acel sistem, sa desfaceti arhiva `gensynsets.zip` si apoi sa fiti capabili sa rulati clasa (Java) `GenSynsets`.

Utilizare

GenSynsets este un utilitar conceput pentru a fi folosit din linia de comanda. Forma generala a liniei de comanda este:

```
java -classpath .\jwordnet.jar GenSynsets -pos
nounadj [-enrich] [-cs charset] [-l SynList]
WnDictPath E_F F_E OutFile
```

unde:

- `-pos` specifica partea de vorbire considerata: `noun` sau `adj`
- `-enrich` este folosit numai in cazul adjectivelor, iar aparitia sa determina aplicarea tehnicii de imbogatire (a se vedea articolul)
- `-cs` seteaza setul de caractere asa cum va apare in fisierul de iesire XML (`<?xml version="1.0" encoding="charset">`). Valoarea prestabilita este **iso-8859-1**.

- `-l SynList`; daca switch-ul este prezent, atunci synset-urile corespunzatoare limbii straine vor fi generate numai pentru synset-urile englezesti ale caror offset-uri apar in lista specificata. Daca switch-ul nu este prezent, atunci toate synset-urile englezesti (corespunzatoare POS-ului precizat) sunt procesate de catre program.
- `WnDictPath` reprezinta calea unde se afla baza de date WordNet (`c:\wn16\dict`).
- `E_F`, `F_E` sunt dictionarele bilingve English-Foreign respectiv Foreign-English.
- `OutFile` va continethe rezultatul procesarii (synset-urile corespunzatoare limbii straine).

Formatul fisierelor

dictionarele bilingve

- dictionarul English-Foreign contine linii, fiecare dintre acestea incluzand cuvantul englezesc, urmat de un spatiu si de traducerile aferente:

```
eword fword1;fword2,fword3;fword4,fword5
```

In scopul de a distinge intre `fword1`, `fword2`, etc. sunt folositi doi separatori. Caracterul punct si virgula separa sensuri diferite ale unui cuvant dat (`eword`). Virgula separa diferitele sinonime referitoare la o anumita semnificatie a cuvantului (`eword`).

- dictionarul Foreign-English contine linii, fiecare dintre acestea incluzand cuvantul din limba straina, urmat de un spatiu si de traducerile in limba engleza corespunzatoare:

```
fword eword1;eword2,eword3;eword4,eword5
```

In scopul de a distinge intre `eword1`, `eword2`, etc. sunt folositi doi separatori. Caracterul punct si virgula separa sensuri diferite ale unui cuvant dat (`fword`). Virgula separa diferitele sinonime referitoare la o anumita semnificatie a cuvantului (`fword`).

fisierul de iesire

Iesirea este furnizata ca fisier in format XML. Astfel, fisierele XML produse de catre **GenSynsets** pot fi usor transformate, prin intermediul XSLT, in alte formate (XML, HTML, etc.) si pot fi utilizate de alte aplicatii. Pentru mai multe detalii a se vedea fisierul DTD (`fsynsets.dtd`) pe care se bazeaza fisierul de iesire XML.

Anexa 2. Exemple de output pentru substantive romanesti

English synset: {banishment, proscription}

the act of banishing someone

e-sets:

eword	e-set	score
proscription	{surghiunire, exilare}	2.0
banishment	{exilare, surghiunire, exil, surghiun, expulzare, ostracizare}	0.8333333

proposed foreign synset(s):

- {exilare, surghiunire, exil, surghiun, expulzare, ostracizare}

English synset: {ostracism}

the act of excluding someone from society by general consent

e-sets:

eword	e-set	score
ostracism	{ostracism}	1.0
ostracism	{ostracizare, surghiunire}	0.5

proposed foreign synset(s):

- {ostracism}

English synset: {substance, matter}

that which has mass and occupies space; "an atom is the smallest indivisible unit of matter"

e-sets:

eword	e-set	score
substance	{materie, substanta}	3.0
matter	{materie, substanta}	3.0
matter	{fond}	3.0
substance	{esenta, fond}	2.0
matter	{chestiune, problema}	1.0
matter	{lucru, fapt, obiect}	1.0
matter	{subiect}	1.0

proposed foreign synset(s):

- {materie, substanta}
 - {materie, substanta, fond}
-

English synset: {gesture}

motion of hands or body to emphasize or help to express a thought or feeling

e-sets:

eword	e-set	score
gesture	{gest}	1.0
gesture	{gest, atitudine}	0.5

proposed foreign synset(s):

- {gest}

English synset: {universe, existence, nature, creation, world, cosmos, macrocosm}

everything that exists anywhere; "they study the evolution of the universe"; "the biggest tree in existence"

e-sets:

Eword	e-set	score
World	{lume}	7.0
Cosmos	{lume}	7.0
Creation	{lume, univers, natura}	5.3333335
Universe	{univers}	5.0
world	{univers, cosmos}	4.5
cosmos	{cosmos, univers}	4.0
macrocosm	{macrocosm, univers}	3.5
universe	{cosmos}	3.0
nature	{fire}	3.0
nature	{natura, fire}	2.0
nature	{caracter, temperament, fire}	1.3333334
nature	{natura}	1.0
nature	{esenta, caracter}	1.0
creation	{creatiune}	1.0
creation	{creatie}	1.0
existence	{existenta, fiintare, vietuire}	0.6666667
creation	{creare, faurire, zamislire}	0.6666667
existence	{prezenta, existenta}	0.5
existence	{fiinta, vietuitoare}	0.5
nature	{natura, specie, fel, gen}	0.5

proposed foreign synset(s):

- {univers, existenta, fiintare, vietuire, fire, lume, natura, macrocosm}

English synset: {advocate, proponent, exponent}

a person who pleads for a cause or idea

e-sets:

eword	e-set	score
advocate	{avocat}	1.0
exponent	{exponent}	1.0
proponent	{sustinator, partizan, adept}	0.6666667
advocate	{aderent, adept, partizan}	0.33333334
exponent	{interpret, indrumator, talmacitor}	0.33333334
exponent	{exponent, factor, reprezentant}	0.33333334

proposed foreign synset(s):

- {avocat, sustinator, partizan, adept, exponent}

English synset: {absolutism, tyranny, despotism}

dominance through threat of punishment and violence

e-sets:

eword	e-set	score
despotism	{despotism}	5.0
tyranny	{tiranie}	3.0
absolutism	{absolutism}	1.0

proposed foreign synset(s):

- {absolutism, tiranie, despotism}

Anexa 3. Exemple de output pentru adjective romanesti

Synset-uri de adjective obtinute fara imbogatire

English synset: {brumous, foggy, hazy, misty}

filled or abounding with fog or mist; "a brumous October morning"

e-sets:

eword	e-set	score
foggy	{cetos, nebulos, neclar, confuz}	3.25
foggy	{cetos, neguros}	3.0
hazy	{incetosat, neguros}	3.0
brumous	{posomorat, cetos, neguros}	2.6666667
misty	{cetos, incetosat, innorat, aburit}	2.0
misty	{vag, neclar, confuz, neinteligibil}	1.5
hazy	{vag, neclar, estompat, sters}	0.5
hazy	{confuz, intunecat, nesigur}	0.33333334

proposed foreign synset(s):

- {posomorat, cetos, neguros, nebulos, neclar, confuz, incetosat, innorat, aburit}

English synset: {cloud-covered, clouded, overcast, sunless}

filled or abounding with clouds

e-sets:

eword	e-set	score
cloud-covered	{innorat, noros}	4.0
clouded	{innorat}	3.0
clouded	{innorat, posomorat}	1.5

proposed foreign synset(s):

- {innorat, noros}
-

English synset: {fair}

free of clouds or rain; "today will be fair and warm"

e-sets:

eword	e-set	score
fair	{bun, frumos}	1.0
fair	{ieftin}	1.0
fair	{balan, balai, blond, deschis}	0.75
fair	{frumos, curat, ingrijit}	0.6666667
fair	{drept, nepartinitor, impartial}	0.6666667
fair	{bun, frumos, placut, prielnic, favorabil}	0.6
fair	{cinstit, onest}	0.5
fair	{cinstit, deschis}	0.5
fair	{convenabil, acceptabil, accesibil, rezonabil}	0.5
fair	{bun, natural, firesc}	0.33333334
fair	{frumos, minunat, atragator, dragut}	0.25

proposed foreign synset(s):

- {bun, frumos}
- {ieftin}

Synset de adjective obtinut cu imbogatire

English synset: {fair}

free of clouds or rain; "today will be fair and warm"

e-sets:

eword	e-set	score
fair	{senin}	2.0
fair	{limpede}	2.0
clear	{clar, curat, luminos, limpede, senin}	1.4
fair	{frumos, curat, ingrijit}	1.3333334
fair	{citet, clar}	1.0
fair	{bun, frumos}	1.0
fair	{ieftin}	1.0
clear	{limpede, lamurit, inteligibil, clar, deslusit}	1.0
clear	{clar, perceptibil, lamurit, limpede, deslusit}	0.8
fair	{balan, balai, blond, deschis}	0.75
clear	{curat, neincarcata, negrevat, integ}	0.75
fair	{drept, nepartinitor, impartial}	0.6666667
fair	{bun, frumos, placut, prielnic, favorabil}	0.6
fair	{cinstit, onest}	0.5
fair	{cinstit, deschis}	0.5
fair	{convenabil, acceptabil, accesibil, rezonabil}	0.5
clear	{liber, deschis}	0.5
clear	{clar, patrunzator}	0.5
fair	{bun, natural, firesc}	0.33333334
fair	{frumos, minunat, atragator, dragut}	0.25

proposed foreign synset(s):

- {senin, clar, curat, luminos, limpede}

Unele comentarii lingvistice privind rezultatele obținute

Theodor Hristea

Dorim să menționăm, încă de la început, faptul că, în majoritatea cazurilor, programele de calculator care implementează algoritmii referitori la WordNet lucrează în mod corect și că, atunci când rezultatele obținute nu sunt cele mai bune cu putință, acest lucru se datorează, în primul rând, imperfecțiunii dicționarului bilingve existente. În pagina de web a proiectului sunt arătate, în mod special, acele situații în care programul nu lucrează corect sau în care propune mai mult de un singur synset românesc, lăsând la latitudinea lingvistului alegerea celui adecvat, în special pe baza glosei. În cele ce urmează, vom încerca să comentăm principalele tipuri de greșeli care pot interveni în urma prelucrării automate și să analizăm cauzele care le-au determinat.

Dorim, în mod special, să semnalăm următoarele trei tipuri de situații: cele în care programul a generat mai multe synset-uri românești dintre care unul este corect, cele în care nu a generat nici un synset românesc și cele, mult mai rare, în care a generat unul sau mai multe synset-uri, dar care sunt greșite.

În acele cazuri în care, pentru un synset englezesc dat, au fost obținute mai multe synset-uri românești posibile, alegerea celui corect (în funcție de glosă) a fost, de cele mai multe ori, evidentă pentru lingvist.

Considerăm că mai interesante au fost situațiile în care, prin program, nu a fost generat nici un synset românesc. Cel mai adesea, cauza o constituie imperfecțiunea dicționarului bilingve, care nu includ cuvintele respective. Alteori doar unul dintre dicționare este de vină, cel mai adesea fiind vorba de dicționarul român-englez, relativ sărac în privința numărului cuvintelor-titlu, dar și în privința celor englezești luate în considerație. În multe dintre situațiile datorate acestui fapt, prin algoritmul propus, se vor obține numai mulțimi elementare neetichetate. În acest caz, nu se generează, conform algoritmului, nici un synset românesc.

Există, prin urmare, situații de naturi diferite în care nu este generat nici un synset românesc. Fie cuvântul nu a fost găsit în dicționarul englez-român, fapt care afectează în mod direct synset-urile englezești cu un singur cuvânt, suficient de frecvente în WordNet, fie a fost găsit, dar, corespunzător lui, au fost obținute numai mulțimi elementare neetichetate. Aceasta din urmă este situația cea mai frecventă. Este, de pildă, cazul lui **crook**, cu sensul "a long staff with one end being hook shaped", ori cazul lui **wreckage**, cu sensul "the remains of something that has been wrecked".

Uneori synset-ul românesc generat de program va fi incorect format datorită funcției de evaluare implementate. Noi funcții de evaluare ar trebui implementate și testate în viitor. Cel mai adesea însă, funcția de evaluare luată în considerație nu lucrează corect tot datorită imperfecțiunii dicționarilor bilingve existente. Este cazul synset-ului format din unicul cuvânt **rule** cu sensul "directions that define the way a game or sport is to be conducted", tradus în românește prin [riglă], precum și al synset-ului format din cuvântul **convention** cu sensul din diplomatie "an international agreement" tradus prin [adunare, întrunire, congres], deci cu sensul de *congress*, în loc de [convenție, acord, contract, învoială, înțelegere, pact, tratat]. Așa cum am mai arătat, este destul de frecventă situația în care un cuvânt românesc care apare în dicționarul englez-român nu se regăsește în dicționarul român-englez. Este, în special, cazul substantivelor provenite din verbe și care au semnificația "acțiunea de a...". Cuvinte importante și uzuale în limba română, cum ar fi **organizare** (de la *a organiza - to organize*) sau **respingere** (de la *a respinge - to reject*), apar ca traduceri ale unor cuvinte englezești, dar nu se regăsesc în dicționarul român-englez. Acest lucru poate duce la eșecul algoritmului de evaluare a e-mulțimilor, întrucât neregăsirea unui cuvânt în dicționarul român-englez aduce cu sine o valoare mai mică a acelei e-mulțimi.

Tot datorită incompletitudinii dicționarilor existente, foarte multe împrumuturi recente existente în limba română (în special în mass-media) nu vor apărea în synset-urile românești generate.

În acele situații în care dicționarul român-englez nu este de vină, cauza erorilor pe care le face programul este de cu totul altă natură și trebuie căutată în sfera conceptelor. Trebuie avut în vedere faptul că limba engleză și, în mod special, engleza americană, la care se referă WordNet, este o limbă incomparabil mai bogată decât limba română. Statistic vorbind, în timp ce româna are maximum 150.000 de cuvinte, engleza americană are aproximativ 450.000 cuvinte (juducând după informațiile furnizate de către lexicograful St. Berg Flexner). Dar, comparativ cu limba română, engleza este o limbă mult mai evoluată nu numai din punct de vedere gramatical și lexical (adică sub raport strict cantitativ), ceea ce înseamnă mai multe cuvinte sau unități lexicale. Ea este totodată mult mai evoluată și sub raport strict semantic, același cuvânt englezesc având adeseori un conținut semantic mult mai bogat decât cuvântul românesc corespunzător. Numeroase cuvinte existente atât în română, cât și în engleză, sunt mai polisemantice în engleză decât în română. Cu alte cuvinte, polisemia cuvintelor englezești este superioară polisemiei cuvintelor românești. Spre exemplu, cuvântului englezesc **feature** cu sensul de "an article of merchandise that is displayed or advertised more than other articles" nu îi corespunde în limba română un cuvânt românesc cu același sens. Suntem deci obligați să recurgem la traducere printr-un grup de cuvinte (o glosă), iar synset-ului englezesc format din singurul cuvânt **feature** care se referă la acest concept nu îi corespunde un synset românesc. În acest caz, programul a lucrat greșit. Este, din nou, o situație care afectează în special synset-urile englezești alcătuite dintr-un singur cuvânt. Un alt exemplu de cuvânt polisemantic englezesc este **foundation**, care ne atrage atenția

prin unul dintre sensurile sale: "a woman's undergarment worn to give shape to the contours of the body". Acest sens al lui **foundation** nu există în română, iar conceptul la care se referă synset-ul conținând unicul cuvânt **foundation** cu acest sens trebuie explicat în limba română prin intermediul unei glose. Lui nu trebuie să îi corespundă nici un synset românesc. Programul a lucrat din nou greșit în această situație, ca și în cazul englezescului **quiver** cu sensul "a case for holding arrows".

O altă situație în care programul nu lucrează corect se referă la anumite substantive englezești folosite cu negație, cum ar fi **matter** cu negație, ca în exemplul "they were friends and it was no matter who won the game". Astfel de substantive se traduc în limba română printr-o colocație al cărei centru îl constituie un substantiv care nu figurează în dicționarul englez-român printre posibilele traduceri ale lui **matter** sau care figurează în dicționar printr-un echivalent de tip locuțional, care nu va fi folosit de algoritmul pe care îl implementează programul. În acest caz programul nu poate găsi synset-ul românesc (sau, în general, străin) corect. În particular, în cazul lui **matter** folosit cu negație, au fost găsite câteva synset-uri românești posibile, dar nici unul nu este corect deoarece nu conține substantivul **importantă (importance)**, care intră în colocația românească corespunzătoare acestui sens. Expresia românească corespunzătoare este un calc după franțuzescul *avoir de l'importance*. Cazurile de calc după franceză sunt foarte frecvente în limba română. Iată de ce simțim nevoia ca programele ulterioare să ia în considerație colocațiile, atât în limba engleză, cât și în limba țintă (în particular, româna).

Alteori, singurul cuvânt din synset-ul englezesc considerat nu se traduce în românește printr-o colocație, ci chiar printr-un cuvânt cu aceeași formă. Și totuși programul nu lucrează corect în unele dintre aceste situații. Este cazul synset-ului alcătuit din unicul cuvânt englezesc **act** și denotând conceptul "lipsă de sinceritate". El a fost tradus greșit în limba română prin synset-ul românesc [**faptă, fapt, act, acțiune**], ce conține, printre altele, un cuvânt românesc cu aceeași formă - **act**. Dar acest sens al englezescului **act** - lipsă de sinceritate - nu există în română. Este un exemplu de ceea ce lingviștii numesc "falși prieteni". În astfel de cazuri este vorba despre cuvinte englezești care sub aceeași formă sau o formă similară există și în alte limbi, dar fără a avea sensul specific limbii engleze. Aceeași situație se întâlnește și în cazul synset-ului [**pattern**] cu sensul inexistent în română "the path that is prescribed for an airplane that is preparing to land at an airport" sau al synset-ului [**cosmos**] cu sensul "any of various mostly Mexican herbs of the genus Cosmos". Multe dintre aceste sensuri sunt specifice englezii americane. Un alt exemplu îl constituie synset-ul [**circumstances**] cu sensul "the state (usually personal) with regard to wealth" tradus greșit prin [**împrejurări, circumstanțe, condiții**]. Sensul din WordNet al lui **circumstances** (plural) există atât în engleza americană, cât și în cea britanică, dar nu și în limba română.

O altă sursă de dificultăți o constituie substantivele la plural. Astfel, există în synset-urile englezești substantive la singular care se traduc prin substantive românești la plural. Exemple din această categorie sunt **foundation** tradus prin

fonduri sau **knowledge** tradus prin **cunoștințe**. Pentru rezolvarea unor astfel de situații au fost incluse în dicționarul român-englez și aceste substantive la plural, dând astfel posibilitatea programului să ia în considerație și mulțimile elementare care conțin substantive la plural.

În limba română, ca și în alte limbi, cum ar fi de exemplu franceza, relația dintre omonimie și polisemie este o problemă extrem de complicată, asupra căreia specialiștii nu au căzut încă de acord. Ea nu constituie o problemă rezolvată. În foarte multe cazuri, după unii cercetători avem de-a face cu două, trei sau chiar mai multe cuvinte omonime, iar după alții cu un singur cuvânt polisemic, care are două, trei sau mai multe sensuri fundamentale, mai mult sau mai puțin înrudite. Un exemplu ar fi cuvântul **bun** (**good**), care în limba română este în primul rând adjectiv, având șapte sensuri fundamentale. În al doilea rând este substantiv cu două pluraluri diferite, dar specializate semantic. Substantivul **bun** cu pluralul **bunuri** are patru sensuri, iar substantivul **bun** cu pluralul **buni** are un singur sens, acela de bunic. Astfel de situații sunt suficient de frecvente în limba română. Programele de calculator concepute în cadrul acestui proiect vor lucra mai bine atunci când se folosesc dicționare care tratează posibilele omonime, în special omonimele așa-zise semantice, ca pe un singur cuvânt polisemantic. Altfel, ar trebui luată de la început în considerație glosa pentru stabilirea sensului, adică a conceptului la care se referă synset-ul englezesc de tradus.

În concluzie, putem spune că principalele probleme care apar în traducerea automată a synset-urilor englezești și care pot genera situații în care programul nu lucrează corect, sunt reprezentate de așa-numiții falși prieteni, de colocații, de calcul lingvistic și de superioritatea polisemiei unor cuvinte englezești în raport cu corespondentele lor românești. Tot ca o concluzie vom observa faptul că cele mai multe probleme au apărut acolo unde synset-ul englezesc era compus dintr-un singur cuvânt, algoritmul neputând decide între sensuri. În viitor se impune, probabil, o tratare diferită a acestor synset-uri. În același timp, trebuie să accentuăm faptul că lipsa unor instrumente de lucru foarte performante (cu referire la dicționare) nu poate decât să îngreuneze o reală evaluare a programelor folosite în traducere.

Nu putem încheia acest scurt comentariu fără a sublinia și unele dintre meritele algoritmilor implementați. Spre exemplu, vom nota faptul că, în ciuda dificultăților menționate, există suficiente exemple de synset-uri cu un singur cuvânt englezesc polisemantic care au fost corect traduse prin synset-uri românești constând, de asemenea, dintr-un unic cuvânt polisemantic. Exemple sunt: synset-ul [**art**], corect tradus prin [**artă**] ori synset-ul [**creation**] corect tradus prin [**creație**].

După cum se știe, conceptele sunt dependente de limbă. În multe cazuri se întâmplă ca un cuvânt englezesc să se refere la un concept foarte larg, iar în română să îi corespundă mai multe cuvinte care se referă la concepte înrudite, dar mult mai specializate. Unul dintre exemplele comentate în acest sens de către Nikolov și Petrova (2001) relativ la bulgară este valabil și pentru limba română,

ilustrând acest fenomen. Este vorba despre synset-ul alcătuit din unicul cuvânt **castle**. În traducerea lui **castle** în limba română apar și cuvinte ca **fortăreață** (**fortress**) sau **citadelă** (**citadel**). Acestea sunt concepte înrudite, dar diferite. Am vrea să ducem comentariul mai departe remarcând faptul că aceasta este o situație în care algoritmul va produce mulțimi neetichetate, iar programele de calculator vor lucra corect prin îndepărtarea acestor mulțimi.

În fine, nu putem să nu remarcăm faptul că, atunci când dicționarele bilingve utilizate au fost corecte și complete, algoritmul implementat s-a dovedit a fi extrem de performant. Astfel, în cazul unor concepte foarte apropiate în limba engleză, diferența fină a fost sesizată de program, care o menține în mod corect în traducerea românească. Este, de pildă, cazul synset-urilor englezești [**banishment**, **proscription**] cu sensul "the act of banishing someone" și respectiv [**ostracism**] cu sensul "the act of excluding someone from society by general consent". Primul a fost tradus în limba română prin [**exilare**, **surghiunire**, **exil**, **surghiun**, **expulzare**, **ostracizare**], în timp ce al doilea a fost tradus prin unicul [**ostracism**]. Românescul **ostracism** este singurul care se referă la consens în luarea deciziei de a ostraciza și, prin folosirea lui în cazul celui de-al doilea synset, este pusă în evidență diferența dintre cele două concepte diferite, dar înrudite la care se referă synset-urile englezești inițiale.

În concluzie, vom spune că un asemenea studiu asupra posibilității generării semiautomate a synset-urilor românești este binevenit și se anunță suficient de promițător. Încurajăm continuarea lui în cazul limbii române, cu lărgirea sferei de investigație, în special prin luarea în considerație, în viitorul apropiat, a colocațiilor. Acestea trebuie avute în vedere atât în limba engleză, cât și în limba română și, mai general, în limba țintă.

Buletinul RORIC-LING

Iunile 7 - 12

Au fost puse **85** de intrebari de catre utilizatori provenind in special din unitatile de invatamant romanesti, dar nu numai. Firmele de software atat din Romania, cat si din strainatate sunt, de asemenea, reprezentate. Unele dintre intrebari au fost puse de mai multe ori, asa cum va fi specificat in buletin. Exista patru mari categorii de intrebari, corespunzand tematicii partii romanesti a proiectului BALRIC-LING, dupa cum urmeaza:

- intrebari cu caracter general;
- intrebari referitoare la programul GenSynsets;
- intrebari cu privire la implementarea RORIC-LING a algoritmilor referitori la WN;
- intrebari generale referitoare la WordNet.

In interiorul buletinului, intrebarile au fost grupate in conformitate cu tematica la care se refera (si nu in ordinea subscrierii, adica in ordine cronologica). Toate informatiile referitoare la numele si datele personale ale utilizatorilor exista in fisierele RORIC-LING, dar au fost sterse din buletin, pentru a facilita citirea si folosirea acestui material, precum si cautarea in cadrul lui dupa tematica.

Cateva statistici:

Intrebari: 85

Tara: **Romania Alta***

Intrebari: 67 18

**Limba
Materna:** **Romana Alta**

Intrebari: 80 5

**Domeniu de
Activitate:** **Educatie Cercetare Industrie Software Altul**

Intrebari: 41 16 16 12

* AUSTRALIA, AUSTRIA, FRANCE, GERMANY, ITALY, UNITED KINGDOM, UNITED STATES

Intrebari cu caracter general

Ati putea sa imi recomandati cateva publicatii relevante privitoare la WordNet?

Cei care nu sunt familiarizati cu WordNet ar trebui sa citeasca "Five Papers on WordNet", articole disponibile in format PostScript si Acrobat (PDF) pe web. Va rugam sa cautati la adresa <http://www.cogsci.princeton.edu/~wn/> sub "Publications".

Daca sunteti deja familiarizati cu WordNet, va recomandam cartea "WordNet: An Electronic Lexical Database", care este acum disponibila la MIT Press. Cartea include articole care descriu arhitectura si continutul WordNet (o actualizare la "Five Papers on WordNet", precum si articole despre cercetarile efectuate pe baza WordNet in domeniile lingvisticii, regasirii informatiei, dezambiguizarii sensului cuvintelor, construirii concordantei semantice, analizei textelor si ingineriei limbajului). Atat cartea, cat si CD-ul aferent pot fi cumparate direct de la MIT Press.

Unde pot gasi documentatie XML profesionala in limba romana?

Nu cunoastem nimic referitor la existenta pe web a documentatiei XML gratuite in limba romana. Va recomandam insa o carte foarte buna, tradusa in limba romana:

Lee Anne Phillips, XML. Editura Teora, 2001

Am facut download la pachetul WordNet pentru PC-uri, dar nu stiu sa il instalez. Ma puteti ajuta, va rog?

Ar fi trebuit sa executati download-ul unui fisier numit "wn16pc.exe". Daca download-ul s-a facut in mod corect, atunci ar trebui sa fie suficienta executia unui dublu-clic pe acest fisier pentru ca el sa se autoextraga. Dupa aceea trebuie sa urmati instructiunile incluse in fisierul INSTALL.txt, pentru a instala efectiv pachetul WordNet.

De unde pot obtine manuale de WordNet?

Va rugam sa va uitati la adresa <http://www.cogsci.princeton.edu/~wn/doc.shtml> Ar trebui sa gasiti acolo o lista de manuale referitoare la WordNet, disponibile online.

Iata cateva proiecte legate de WordNet:

- Retele semantice - in alte limbi decat engleza
- Interfete web
- Interfete locale - necesita downloadul unor fisiere
 - .NET
 - C++
 - COM
 - dBase/MySQL
 - Java
 - Lisp
 - Palm
 - Perl
 - Prolog
 - Python
- Extensii - care extind caracteristicile WordNet sau care o integreaza in sisteme mai ample.

Informatii suplimentare referitoare la aceste proiecte pot fi gasite la adresa <http://www.cogsci.princeton.edu/~wn/links.shtml>

Ce este "Global WordNet Association"?

"Global WordNet Association" este o organizatie publica, necomerciala si cu acces gratuit , care furnizeaza o platforma pentru discutii, precum si pentru distribuirea si conectarea bazelor de date de tip WordNet corespunzatoare tuturor limbilor din lume.

Intrebari referitoare la programul GenSynsets

Va rugam sa explicati de ce output-ul final este generat in format XML.

In acest caz formatul XML faciliteaza prezentarea pe web a rezultatelor programului si permite accesul mai rapid la informatii. XML (the Extensible Markup Language) a fost mai intai ratificat de catre consortiul W3C (World Wide Web Consortium) ca reprezentand standardul pentru schimbul de informatie pe Internet in februarie 1998. XML specifica o modalitate riguroasa, bazata pe text de

a reprezenta structura intrinseca a datelor astfel incat aceasta sa poata fi interpretata in mod neambiguu.

Care este setul de caractere implicit folosit de program (atunci cand parametrul CS nu este specificat in linia de comanda)?

Setul de caractere implicit folosit de program (atunci cand parametrul CS nu este specificat in linia de comanda) este iso-8859-1.

Cum realizati (ce instructiuni Java sunt folosite) operatiile de tip I/O asupra fisierelor si cum se fac citirea/scrierea datelor in diverse limbi?

Operatiile de tip I/O asupra fisierelor care permit lucrul cu date in diverse limbi (codificate cu diverse seturi de caractere) au fost scrise cu utilizarea specificatiilor Java `InputStreamReader`, `BufferedReader`, precum si `OutputStreamWriter` si respectiv `BufferedWriter`.

Ce functii pentru lucrul cu siruri de caracter sunt utilizate in program?

Programul foloseste atat functii Java pentru lucrul cu siruri de caractere, precum si metode definite de utilizator pentru separarea liniilor de text citite din dictionare si pentru extragerea subsirurilor cu semnificatie relativa la algoritm (cum ar fi traducerile cuvintelor.)

Ce clase Java sunt definite in program si ce contin acestea?

Programul defineste o singura clasa, **GenSynsets**, ce cuprinde toate variabilele si metodele necesare implementarii algoritmilor pe care se bazeaza. Dintre metodele definite mentionam pe cele referitoare la operatii I/O cu fisierele dictionarelor, de etichetare a e-set-urilor generate, metoda ce implementeaza strategia de tip backtracking, operatii cu siruri de caractere, sortare etc.

Care este modalitatea pe care se bazeaza, in program, aflarea glosei unui synset englezesc?

Gasirea glosei unui synset se face utilizand metoda `getGloss()` din clasa `Synset`.

Cum sunt determinate (in program) synset-urile finale corespunzatoare limbii straine alese? (pusa de doua ori)

Pentru determinarea synset-urilor finale (corespunzatoare limbii straine alese), programul combina **e-set-urile** etichetate cu valoarea maxima pentru fiecare **eword** tradus, eliminand duplicatele. Aceasta combinatie este implementata printr-o metoda de tip **backtracking**.

Cum trateaza programul cazul in care exista cuvinte ale unui synset, dat prin offset-ul sau, care nu au intrari corespunzatoare in dictionarul English-Foreign (Language)? (pusa de doua ori)

In situatia respectiva sunt utilizate numai cuvintele din synset-uri pentru care exista traducere (au intrari corespunzatoare in dictionarul English-Foreign (Language)).

Ce tehnici de programare sunt utilizate in program? (pusa de doua ori)

Dintre tehnicile de programare folosite trebuie remarcata metoda **backtracking**, prin intermediul careia sunt generate **fsynset-urile**, pornind de la **e-set-uri**, cu eliminarea duplicatelor.

Sunt folosite rezultate ale unor proiecte anterioare de WordNet implementate in Java? (pusa de doua ori)

Programul GenSynsets foloseste rezultatele obtinute in cadrul proiectului JWordNet. JWordNet reprezinta o interfata de sine statatoare, orientata obiect, scrisa in Java, ce implementeaza diversele entitati lexicale si semantice din WordNet. Ea este gandita pentru programatorii Java care doresc sa scrie aplicatii Java portabile, care folosesc WordNet-ul si utilizeaza o copie locala a fisierelor acestuia, sau pentru aceia care considera interfata 'object-oriented' JWordNet preferabila interfetei procedurale existente, scrisa in limbajul 'C'. JWordNet contine clasa de tip generator, DictionaryDatabase, clasele de entitati, IndexWord, Synset, Word si Pointer, precum si clasele de tip enumerare, POS si PointerType.

Cum realizeaza programul parcurgerea tuturor synset-urilor din WordNet-ul englezesc? (pusa de doua ori)

In acest caz, parcurgerea tuturor synset-urilor din WordNet se realizeaza pe baza unei enumerari si a metodei **synsets** apartinand clasei **DictionaryDatabase** (JwordNet).

In ce mod se determina in program cuvintele unui synset precizat prin offset-ul sau? (pusa de doua ori)

In cazul executiei programului pe baza unei liste de synset-uri (date prin offset-urile acestora) se transmit metodei doua argumente: POS (part of speech) si offset-ul unui synset. Gasirea synset-urilor din WordNet-ul englezesc se face pe baza metodei `getSynsetAt()` ce apartine clasei `FileBackedDictionary` (`JWordNet`).

Intrebari cu privire la implementarea RORIC-LING a algoritmilor referitori la WN

De ce nu exista nici un fel de numere in codificarea realizata de dvs. pentru cluster-ele de adjective romanesti? (pusa de doua ori)

Deoarece, pentru simplitate, in cadrul acestui demo am lucrat cu synset-uri curatate si combinate (a se vedea materialul explicativ de pe web). In cadrul unei asemenea implementari am renuntat la o serie de parametri, cum ar fi numarul sensului. Algoritmul existent poate fi usor modificat astfel incat sa poata lucra cu synset-uri in care acest parametru exista si pentru a-l putea lua in considerare. De asemenea, trebuie avut in vedere faptul ca, pentru a putea genera intr-o limba straina cluster-e de adjective de tip WordNet in intregime codificate, trebuie mai intai obtinute toate synset-urile de adjective din acea limba. Numai in acest fel pot fi vazute toate sensurile unui anumit adjectiv, existente in limba tinta. Acest lucru nu a fost inca realizat pentru limba romana, datorita dictionarelor bilingve in format electronic incomplete. Pe de alta parte, noi nu am urmarit generarea unor cluster-e de adjective identice cu cele din WordNet-ul american, ci numai generarea unor synset-uri si cluster-e de adjective "de tip WN". Programele existente pot fi inasa usor modificate pentru a genera exact aceeasi forma a cluster-elor de adjective existenta in WordNet.

Dupa cum se stie, multe adjective sunt limitate la pozitile sintactice pe care le pot ocupa. Este aceasta limitare codificata in WN si cum? Ati realizat aceasta codificare in propria dumneavoastra implementare? (pusa de doua ori)

Asa cum s-a mentionat deja, multe adjective sunt limitate la pozitile sintactice pe care le pot ocupa, iar aceasta limitare este de obicei codificata in WordNet. Intrucat aceasta este o limitare care se refera la forma cuvintului, ea este codificata pentru adjective individuale si nu pentru intregi synset-uri. Considerati, spre exemplu, cluster-ul `awake/asleep`, avand ambele adjective limitate la pozitia de predicat. Desi acestea sunt cuvintele-cap ale cluster-ului, limitarea respectiva nu este

valabila pentru toate sinonimele din cluster. De aceea, cuvintele individuale limitate in acest fel sunt toate codificate cu (p). In cazul adjectivelor limitate la pozitii de atribut codul este (a). In fine, pentru cele cateva adjective care pot sa apara numai imediat dupa un substantiv, codul este (ip), de la "postnominal imediat" ("immediately postnominal"). Aceasta codificare nu a fost inca implementata pentru cluster-ele de adjective romanesti.

De ce nu discutati, in egala masura, verbele in WordNet, in cadrul acestui proiect?

Pentru ca aceasta nu este o discutie exhaustiva cu privire la WordNet si/sau generarea automata a unor baze de date de tip WordNet pentru alte limbi decat engleza. Am ales sa discutam cele doua structuri de baza din WordNet - ierarhia si cluster-ul - pentru care am studiat substantivele si adjectivele in WordNet. Pentru o discutie detaliata cu privire la toate chestiunile legate de traducerea WordNet, vezi proiectul BALKANET, la adresa <http://www.ceid.upatras.gr/Balkanet>

Puteti mentiona unele dintre dificultatile intalnite atunci cand ati implementat algoritmi descriși in cazul propriei limbi - romana? (pusa de trei ori)

Principalele dificultati care au intervenit in traducerea automata a synset-urilor englezești in synset-uri romanesti au fost generate de asa-numiti falsi prieteni, de colocatii, de calc, precum si de faptul ca polisemia multor cuvinte englezești este superioara celei a cuvintelor romanesti corespunzatoare. Pentru explicatii detaliate privitoare la toate aceste probleme, va recomandam sa cititi comentariile lingvistice existente in pagina web a proiectului.

Intrebări generale referitoare la WordNet

Ce este asa-numita "matrice lexicala" in WordNet?

Matricea lexicala este o matrice in care formele tip ale cuvintelor sunt imagnate ca reprezentand capetele coloanelor; sensurile cuvintelor reprezinta capete de linii. O intrare intr-o celula a matricii implica faptul ca forma din acea coloana poate fi folosita (intr-un context adecvat) pentru a exprima sensul corespunzator acelei linii. Daca exista doua intrari in aceeasi coloana, cuvantul este polisemantic; daca exista doua intrari in aceeasi linie, atunci cele doua cuvinte sunt sinonime (relativ la un context). Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format

Face WordNet distinctia intre relatii semantice si relatii lexicale?

WordNet face distinctia intre relatii semantice si relatii lexicale. Accentul in WN se pune pe relatii semantice, dar sunt incluse si unele relatii lexicale. Totusi, reseaua WN este organizata in conformitate cu relatii semantice, care sunt indicate prin pointeri.

De unde credeti ca a aparut necesitatea de a se face, in WordNet, partitionarea in substantive, verbe, adjective si adverbe?

Sinonimia este relatia centrala in WordNet. Definirea sinonimiei in termeni de substituibilitate face necesara partitionarea WN in substantive, verbe, adjective si adverbe.

Asa cum se remarca in "Five Papers on WordNet", "daca conceptele sunt reprezentate prin synset-uri si daca sinonimele trebuie sa fie interschimbabile, atunci cuvinte apartinand unor categorii sintactice diferite nu pot fi sinonime (nu pot forma synset-uri) deoarece nu sunt interschimbabile. Substantivele exprima concepte nominale, verbele exprima concepte verbale si elementele modificatoare furnizeaza modalitati de a califica aceste concepte. Cu alte cuvinte, folosirea synset-urilor pentru reprezentarea intelesurilor cuvintelor este consistenta cu probele psiholingvistice referitoare la faptul ca substantivele, verbele si elementele modificatoare sunt organizate in mod independent in memoria semantica."

Sinonimia si antonimia sunt relatii lexicale intre forme ale cuvintelor. Dar hiponimia si hiperonimia?

Spre deosebire de sinonimie si antonimie, care sunt relatii lexicale, hiponimia/hiperonimia (numite, de asemenea, subordonare / supraordonare, submultime/supramultime sau relatia ISA) este o relatie semantica intre intelesuri ale cuvintelor.

Hiponimia este tranzitiva si asimetrica (Lyons, 1977, vol.1) si, intrucat, de regula, exista un singur supraordonat, ea genereaza o structura semantica ierarhica, in care se spune despre un hiponim ca este dedesubtul elementului sau supraordonat. Astfel de reprezentari ierarhice sunt folosite pe scara larga in constructia sistemelor de regasire a informatiei, unde se numesc sisteme bazate pe mostenire (Touretzky, 1986): un hiponim mosteneste toate trasaturile conceptului generic si adauga cel putin o caracteristica care il distinge de elementul sau supraordonat si de orice alte

hiponime ale aceluia supraordonat. Aceasta conventie furnizeaza principiul central de organizare a substantivelor in WN.

Va rog sa imi explicati pe scurt relatia de meronimie si sa indicati unde este ea prezenta in WordNet.

Una dintre relatiile semantice din WordNet este relatia parte - intreg (sau relatia HASA), cunoscuta specialistilor ca meronimie / holonimie. Relatia meronimica este tranzitiva (cu calificari) si asimetrica (Cruse, 1986). Ea poate fi utilizata pentru a construi o ierarhie de parti (cu unele rezerve, intrucat un meronim poate avea multe holonime). Se presupune ca asa-numitul concept de "parte a unui intreg" poate fi parte a unui concept al intregului, desi se recunoaste faptul ca implicatiile acestei presupuneri merita o discutie mai atenta decat cea care le este rezervata in acest cadru de lucru. Meronimia este prezenta in WN in organizarea synset-urilor de substantive.

Exista in WordNet si relatii morfologice?

Asa cum se mentioneaza in "Five Papers on WordNet", "o clasa importanta de relatii lexicale o constituie relatiile morfologice dintre formele flexionare ale cuvintelor. Initial, interesul a fost limitat la relatii semantice; nu au fost facute planuri pentru includerea in WN a relatiilor morfologice. Pe masura ce lucrurile au progresat insa, a devenit din ce in ce mai evident faptul ca, daca WordNet avea sa fie utila cuiva din punct de vedere practic, ea va trebui sa trateze si morfologia flexionara. Spre exemplu, daca cineva avea sa plaseze cursorul pe cuvantul **trees** si sa ceara informatii, WordNet nu trebuia sa raspunda ca acest cuvint nu se afla in baza de date. Era necesar un program care sa elimine sufixul de plural si care apoi sa caute **tree**, care in mod sigur exista in baza de date. Aceasta cerinta a condus la dezvoltarea unui program care sa se ocupe de morfologia flexionara."

Care este principala diferenta intre o definitie standard dintr-un dictionar obisnuit si o definitie din WordNet, referitoare la substantive, spre exemplu?

Definitia standard "tinteste" in sus, spre un termen supraordonat, nu lateral, inspre termeni aflati in relatie de coordonare sau in jos, catre hiponime. Spre exemplu, definitia cuvintului **tree** in dictionarele standard trimite la termenul supraordonat **planta**, dar nu contine nici o informatie despre termeni coordonati. O definitie dintr-un dictionar standard subliniaza cateva deosebiri importante si ii reaminteste cititorului de ceva presupus deja cunoscut; destinatia ei nu este aceea de a reprezenta un catalog de cunostinte generale.

Ce imi puteti spune despre relatia semantica numita relatie de tip ISA si despre implementarea ei referitor la substantive in WordNet?

Relatia semantica care este reprezentata in WordNet prin '@->' a fost numita relatia ISA sau relatie hiperonimica sau supraordonata (intrucat trimite la un hiperonim sau termen supraordonat). Ea se deplaseaza de la specific la generic si, prin urmare, reprezinta o generalizare. Relatia semantica inversa '~>' merge de la generic la specific (de la supraordonat la hiponim) si, prin urmare, reprezinta o specializare.

Asa cum se observa in "Five Papers on WordNet", "intrucat un substantiv, de regula, are un singur element supraordonat, dictionarele standard il includ pe acesta in definitie; intrucat un substantiv poate avea multe hiponime, dictionarele englezești nu indica lista acestora (dictionarul francezesc "Le Grand Robert" reprezinta o exceptie). Chiar daca relatia de specializare nu este facuta explicita in dictionarele standard ale limbii engleze, ea este un derivat logic al relatiei de generalizare. In WordNet lexicografii codifica relatia de generalizare '@->' in mod explicit, printr-un pointer etichetat intre concepte lexicale sau sensuri. Atunci cand fisierele lexicografilor sunt convertite in mod automat in baza de date lexicale, un pas in acest proces este acela de a insera pointeri inversi corespunzator relatiei de specializare '~>'. Astfel, baza de date lexicale este o ierarhie in care se poate cauta in sus sau in jos cu viteze egale." Informaticienii numesc astfel de ierarhii "sisteme bazate pe mostenire", intrucat ei iau in considerare faptul ca anumite entitati mostenesc proprietati de la elementele lor supraordonate generice. Toate proprietatile elementului supraordonat sunt presupuse a fi, in egala masura, proprietati ale celui subordonat. In loc de a afisa acele proprietati in mod redundant de doua ori, ele sunt mentionate numai impreuna cu elementul supraordonat. Un pointer de la elementul subordonat la cel supraordonat este interpretat ca spunand "pentru proprietati suplimentare, vezi aici".

Se spune ca WordNet este un sistem bazat pe mostenirea lexicale. Va rog sa imi dati un exemplu in cazul substantivelor si sa imi explicati implementarea corespunzatoare din WordNet.

WN este, intr-adevar, un sistem bazat pe mostenirea lexicale. In WN a fost depus un efort sistematic pentru conectarea hiponimelor cu elementele lor supraordonate (si vice versa). In WN, o intrare pentru cuvantul **tree**, spre exemplu, contine o referire sau pointer '@->' catre o intrare corespunzatoare lui **planta**. Pointerul este etichetat "supraordonat" prin intermediul simbolului arbitrar '@'. In baza de date, pointerul '@' catre supraordonatul **planta** va fi reflectat printr-un pointer invers '~' catre **tree** in interiorul synset-ului corespunzator lui **planta**. Acest pointer este etichetat ca hiponim, prin intermediul simbolului arbitrar '~'. Calculatorul este programat astfel incat sa poata folosi acesti pointeri etichetati pentru a construi informatia pe care utilizatorul o solicita la un moment dat. Simbolurile arbitrar '@'

si '~' sunt suprimate atunci cand informatia ceruta este afisata. Synset-ul corespunzator lui **tree** ar arata cam asa:

{tree,plant,@ conifer,~alder,~...}

unde '...' este "umplut" cu multi alti pointeri catre hiponime. Synset-ul corespunzator lui **planta** ar arata cam asa

{plant,flora,organism,@ tree,~...}.

Exista argumente de natura psiholingvistica precum ca memoria lexicala umana referitoare la substantive este un sistem bazat pe mostenire?

Prima persoana care a pretins acest lucru in mod explicit pare a fi fost Quillian (1967, 1968). Testari experimentale ale propunerii lui Quillian au fost comunicate in cadrul unui referat de catre Collins si Quillian (1969). Ambii au presupus ca timpii de reactie pot fi folositi pentru a indica numarul de niveluri ierarhice care separa doua sensuri.

O concluzie reprezentand o alternativa - cea pe care se bazeaza WordNet - este aceea ca presupunerea de mostenire este corecta, dar ca timpii de reactie nu masoara ceea ce Collins si Quillian, ca si altii, au presupus. Este posibil ca timpii de reactie sa indice o distanta pragmatica, mai degraba decat una semantica - o diferenta in utilizarea cuvantului si nu una referitoare la sens (Miller si Charles, 1991).

Toate substantivele sunt incluse intr-o unica ierarhie in WordNet? (pusa de doua ori)

In WN substantivele sunt partitionate cu ajutorul unei multimi relativ mici de concepte generice care au fost selectate ca reprezentand fiecare elementul de inceput unic al unei ierarhii. Aceste ierarhii multiple corespund unor campuri semantice relativ distincte, fiecare avand propriul vocabular.

WN a adoptat urmatoarea multime de 25 de elemente de inceput unice:

{act, action, activity}	{natural object}
{animal, fauna}	{natural phenomenon}
{artifact}	{person, human being}
{attribute, property}	{plant, flora}
{body, corpus}	{possession}
{cognition, knowledge}	{process}

{communication}
{event, happening}
{feeling, emotion}
{food}
{group, collection}
{location, place}
{motive}

{quantity, amount}
{relation}
{shape}
{state, condition}
{substance}
{time}

Cel mai important criteriu in alegerea acestor componente semantice primitive este acela ca, in mod colectiv, ele ar trebui sa furnizeze un loc fiecarui substantiv englezesc. Ierarhiile rezultate variaza mult in dimensiune si nu se exclud reciproc. In ansamblu insa ele acopar domenii lexicale si conceptuale distincte. Ele au fost selectate dupa ce s-au luat in considerare combinatiile posibile de tip substantiv-adjectiv la care ne putem astepta sa intervina in limba engleza. (Aceasta analiza a fost efectuata de catre Philip N. Johnson-Laird).

Ce inseamna "concepte generice" cu referire la substantive in WordNet?

Se spune despre ierarhiile nominale din WN ca ele au un nivel, undeva la mijloc, unde sunt atasate majoritatea trasaturilor distinctive. Acesta este asa-numitul "nivel de baza", iar conceptele nominale de la acest nivel se numesc "categorii ale nivelului de baza" sau "concepte generice" (Berlin, Breedlove si Raven, 1966, 1973). Rosch (1975; Rosch, Mervis, Gray, Johnson si Boyes-Braem, 1976) au extins aceasta generalizare: pentru conceptele aflate la nivelul de baza, pot fi enuntate multe trasaturi caracteristice. Deasupra nivelului de baza, descrierile sunt concise si generale. Dedesubtul nivelului de baza, prea putin mai este adaugat caracteristicilor care deosebesc conceptele de baza.

Credeti ca este posibil sa se identifice sensuri alternative ale unui cuvânt numai prin folosirea sinonimelor? Cum trateaza WordNet aceasta problema? (pusa de doua ori)

Asa cum se remarca in "Five Papers on WordNet", "pe masura ce acoperirea realizata de WN s-a largit, a devenit din ce in ce mai evident faptul ca sensuri alternative ale cuvintelor nu pot fi intotdeauna identificate prin folosirea sinonimelor. Mult mai tarziu, prin urmare, s-a decis includerea trasaturilor distinctive, in acelasi mod in care o fac dictionarele conventionale, prin includerea unor scurte glose explicative, ca o parte a synset-urilor continand cuvinte polisemantice. Acestea sunt marcate fata de restul synset-urilor prin paranteze".

Meronimele constituie caracteristici distinctive pe care hiponimele le mostenesc in WN?

Asa cum se remarca in "Five Papers on WordNet", "meronimele reprezinta caracteristici distinctive pe care hiponimele le pot mosteni. In consecinta, meronimia si hiponimia se intrepatrund in moduri complexe. Spre exemplu, daca **beak** si **wing** sunt meronime ale lui **bird** si daca **canary** este un hiponim al lui **bird**, atunci, prin mostenire, **beak** si **wing** trebuie, de asemenea, sa fie meronime ale lui **canary**".

Partile pot fi hiponime, cat si meronime? Daca da, va rog sa imi dati un exemplu din WordNet.

Conexiunile dintre meronimie si hiponimie sunt complicate de faptul ca partile sunt atat hiponime, cat si meronime. Exemplul care este dat in "Five Papers on WordNet" este synset-ul {**beak, bill, neb**}, care este un hiponim al lui {**mouth, muzzle**}, care, la randul sau, este un meronim al lui {**face, countenance**} si un hiponim al lui {**orifice, opening**}. O problema frecventa care apare in stabilirea relatiei adecvate dintre hiponimie si meronimie se naste dintr-o tendinta generala de a atasa caracteristici aflate prea sus in ierarhie. Spre exemplu, daca **wheel** este gandit ca un meronim al lui **vehicle**, atunci saniile vor mosteni roti pe care nu ar trebui sa le aiba. Intr-adevar, in WN a fost creat un synset special pentru conceptul {**wheeled vehicle**}.

In ce ierarhii din WordNet este cel mai adesea prezenta meronimia?

Meronimele au tendinta sa apara cel mai frecvent in legatura cu cuvinte care denota obiecte fizice. In WN meronimia este gasita in special in ierarhiile {**body, corpus**}, {**artifact**} si {**quantity, amount**}.

Este adevarat ca relatia "parte-din" este tranzitiva?

Relatia "parte-din" este adesea comparata cu relatia "un fel de": ambele sunt asimetrice si (cu unele rezerve) tranzitive si ambele pot face legatura dintre termeni in mod ierarhic (Miller si Johnson-Laird, 1976). Cu alte cuvinte, partile pot avea parti: un deget este o parte a unei maini, o mana este o parte a unui brat, un brat este o parte a unui corp: termenul **finger** (deget) este un meronim al termenului **hand** (mana), **hand** este un meronim al lui **arm** (brat), iar **arm** este un meronim al lui **body** (corp). Dar constructia "parte-din" nu reprezinta intotdeauna un test de incredere al meronimiei. In multe imprejurari, tranzitivitatea pare a fi limitata (Lyons, 1977).

Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format PostScript si Acrobat (PDF). Va rugam sa cautati la adresa <http://www.cogsci.princeton.edu/~wn/> sub "Publications".

Este adevarat ca exista diferite tipuri de relatii de tip "parte-din"? Care este situatia implementarii lor in WordNet?

Winston et al. (1987) diferentiaza sase tipuri de meronime: component-obiect (creanga/copac), membru-colectie (copac/padure), portie-masa (felie/tort), material-obiect (aluminu/avion), caracteristica-activitate (a plati/a cumpara), precum si loc-zona (Princeton/New Jersey). Chaffin, Hermann si Winston (1988) adauga o a saptea: faza-proces (adolescenta/crestere). Meronimia este, in mod evident, o relatie semantica complexa - sau o multime de relatii. Numai trei dintre tipurile de meronimie sunt codificate in WN: "este o componenta parte a", "este membru al" si "este materialul din care este facut". Dintre acestea trei, cea mai frecventa este relatia pe care o putem numi "este o componenta a".

Exista relatia de antonimie si intre substantive? Daca da, cum este ea reprezentata in WordNet? (pusa de doua ori)

Asa cum se remarca in "Five Papers on WordNet", "opozitia semantica nu este o relatie fundamentala in organizarea substantivelor, dar ea exista si deci merita propria reprezentare in WordNet. Spre exemplu, synset-urile pentru **man** si **woman** ar contine

{ [man, woman,!], person,@ ... (a male person) }

{ [woman, man,!], person,@ ... (a female person) }

unde relatia simetrica de antonimie este reprezentata prin pointerul '!', iar parantezele drepte indica faptul ca antonimia este o relatie lexicala intre cuvinte, mai degraba decat o relatie semantica intre concepte".

Care sunt principalele relatii semantice luate in considerare in WordNet cu privire la substantive?

Principalele relatii semantice luate in considerare in WordNet cu privire la substantive sunt hiponimia, meronimia si antonimia. Atunci cand toate aceste trei tipuri de relatii semantice sunt incluse, rezultatul este o retea de substantive extrem de interconectate.

Synset-urile de adjective din WordNet contin numai adjective?

Synset-urile de adjective din WordNet contin in majoritate adjective, dar au fost incluse si unele substantive si grupuri prepozitionale care functioneaza adesea ca elemente modificatoare. Discutia purtata in cadrul RORIC-LING se limiteaza la adjective.

Care sunt principalele clase de adjective care sunt luate in considerare in WordNet?

WordNet imparte adjectivele in doua mari clase: descriptive si relationale. Adjectivele descriptive atribuie substantivului cap valori ale unor attribute tipic bipolare si, in consecinta, sunt organizate in termenii unor opozitii binare (antonimie) si ai similaritatii sensului (sinonimie). Adjectivele descriptive care nu au antonime directe sunt considerate a avea antonime indirecte datorita similaritatii lor semantice cu adjective care au antonime directe. WN contine pointeri intre adjective descriptive care exprima valoarea unui atribut si substantivul prin care acel atribut este lexicalizat. Adjectivele relationale sunt presupuse a reprezenta variante stilistice ale unor substantive cu rol modifier si deci sunt puse in legatura cu fisierile de substantive corespunzatoare. Adjectivele cromatice sunt tratate ca un caz special.

Ce inseamna, in mod exact, un adjectiv descriptiv?

Un adjectiv descriptiv este un adjectiv care atribuie o valoare a unui atribut unui substantiv. Cu alte cuvinte, a spune **x este Adj** inseamna a presupune ca exista un atribut **A** astfel incat $A(x)=Adj$. A spune "Pachetul este greu" inseamna a face presupunerea ca exista un atribut **GREUTATE** astfel incat **GREUTATE(pachet) = greu**. In mod similar, **scund** si **inalt** sunt valori pentru atributul **INALTIME**. WN contine pointeri intre adjective descriptive si synset-urile de substantive care se refera la attributele corespunzatoare.

Se aseamana prin ceva organizarea semantica a adjectivelor descriptive in WordNet cu aceea a substantivelor? (pusa de doua ori)

Organizarea semantica a adjectivelor descriptive este complet diferita de aceea a substantivelor. In cazul adjectivelor nu exista nici o relatie care sa genereze ierarhii nominale. Organizarea semantica a adjectivelor este privita in mod mult mai natural ca reprezentand un hiperspatiu abstract cu N dimensiuni si nu un arbore ierarhic.

Care este relatia semantica de baza dintre adjective in WordNet, cea de antonimie sau cea de similaritate? Cum este ea reprezentata in WordNet? (pusa de doua ori)

Relatia semantica de baza dintre adjective descriptive este antonimia. Importanta antonimei a devenit evidenta mai intai in urma rezultatelor obtinute pe baza testelor de asociere a cuvintelor. Importanta antonimiei in organizarea adjectivelor descriptive devine usor de inteles atunci cand se are in vedere faptul ca functia acestor adjective este aceea de a exprima valori ale atributelor si ca majoritatea atributelor sunt bipolare. Adjectivele antonimice exprima valori opuse ale unui atribut. Spre exemplu, antonimul lui **heavy** (greu) este **light** (usor), care exprima o valoare aflata la polul opus al atributului GREUTATE. In WN aceasta opozitie binara este reprezentata prin pointeri etichetati reciproci: **heavy!->light** si **light!->heavy**.

Poate fi relatia de antonimie atat de importanta avand in vedere faptul ca multe adjective descriptive nu au antonime? (pusa de doua ori)

Intrucat multe adjective descriptive nu au antonime, in WN a fost introdus un pointer de similaritate care este folosit pentru a indica faptul ca adjectivele care nu au antonime sunt similare ca sens cu adjective care au antonime. Gross, Fischer si Miller (1989) propun ca synset-urile de adjective sa fie privite ca niste cluster-e de adjective, asociate prin similaritate semantica cu un adjectiv central ce face legatura dintre cluster si un alt cluter, care prin contrast se afla la polul opus al atributului. Gross, Fischer si Miller fac distinctia intre antonimele directe, cum ar fi **heavy/light** (greu/usor) - care sunt perechi lexicale opuse conceptual - si antonimele indirecte, cum ar fi **heavy/weightless** (greu/fara greutate) - care sunt opuse conceptual fara a reprezenta perechi lexicale. In aceasta formulare, toate adjectivele descriptive au antonime; cele care nu poseda antonime directe au, in schimb, antonime indirecte, i.e. sunt sinonime ale unor adjective avand antonime directe.

Reprezinta organizarea adjectivelor in WordNet o garantie a faptului ca toate adjectivele descriptive au antonime? (pusa de doua ori)

Unele adjective descriptive nu au antonime **directe**. Totusi, in organizarea adjectivelor din WN, cele care nu au antonime directe sunt considerate a avea antonime **indirecte**, i.e. ele sunt sinonime ale unor adjective care poseda antonime directe. In aceasta formulare, toate adjectivele descriptive au antonime.

Cum sunt stabilite in WordNet antonimele indirecte?

În WN acele adjective care nu au antonime directe posedă, în schimb, antonime indirecte, i.e. sunt sinonime ale unor adjective având antonime directe. Antonimele directe sunt reprezentate printr-un pointer de antonimie, '!=>'; antonimele indirecte sunt mostenite prin similaritate, relație indicată prin pointerul de similaritate '&=>'.

Care este, pe scurt, modelul de baza prezentat de autorii WordNet cu privire la adjective?

Modelul de baza prezentat de autorii WordNet cu privire la adjective constă în partitionarea adjectivelor în două mari tipuri, și anume cele descriptive (care intră în cluster-e bazate pe antonimie) și cele relationale (care sunt similare substantivelor utilizate ca modificatori). Fără a pretinde o acoperire completă, autorii WN au convingerea că acest model acoperă majoritatea adjectivelor existente în limba engleză.

Ce știți despre relația de gradualitate și cum a fost ea implementată în WordNet?

Conform lui Cliff (1959), un adjectiv cu grade de comparație poate fi definit ca fiind un adjectiv a cărui valoare poate fi multiplicată prin intermediul adverbelor de comparație, cum ar fi foarte, oarecum etc. (în engl. very, decidedly, intensely, rather, quite, somewhat, pretty, extremely).

Gradarea (comparația) mai trebuie să fie privită și ca o relație semantică care organizează memoria lexicală în cazul adjectivelor (Bierwisch, 1989). Pentru unele atribute ea poate fi exprimată prin intermediul sirurilor ordonate de adjective, toate adjectivele din sir "tintind" către același substantiv din WN care denota atributul.

Așa cum se remarcă în "Five Papers on WordNet", "reprezentarea relațiilor ordonate prin pointeri etichetați între synset-uri nu ar fi dificilă, dar s-a estimat că dintre cele peste 2500 cluster-e de adjective nu mai mult de 2% ar putea fi organizate în acest fel. Întrucât relația de gradualitate, importantă din punct de vedere conceptual, nu joacă un rol central în organizarea adjectivelor, ea nu a fost codificată în WordNet".

Exista vreo legatura in WordNet intre substantivul exprimand un atribut si adjectivul exprimand valori ale acelui atribut? (pusa de doua ori)

Substantivul care denota atributul (de ex. LENGTH - lungime) si toate adjectivele exprimand valori ale acelui atribut (in acest caz long, short, lengthy etc.) sunt legate in WN printr-un pointer.

Cum sunt introduse in WordNet denumirile de culori?

In WN opozitia **colored/colorless** (colorat/incolor) este utilizata pentru a introduce denumirile culorilor. Nuantele sunt codificate in mod similar culorilor, iar nuantele de gri (de la alb la negru) sunt codificate ca similare lui **gray** (gri), care apartine unui cluster tripartit impreuna cu alb si negru, furnizand un continuum gradat.

Ce sunt adjectivele relationale?

Adjectivele relationale, care au fost discutate pe larg mai intai de catre Levi (1978), inseamna ceva precum "al, in legatura cu sau asociat cu" un anumit substantiv si joaca un rol asemanator cu cel al substantivelor avand functie de modifier. (Spre exemplu, **dentar** din **igiena dentara** este asociat lui **dinte**).

Care sunt principalele diferente dintre adjectivele relationale si adjectivele descriptive? (pusa de doua ori)

Principalele diferente sunt urmatoarele:

1. Adjectivele relationale difera de adjectivele descriptive prin aceea ca ele nu se raporteaza la un atribut.
2. Adjectivele relationale nu se refera la o proprietate a substantivului cap corespunzator.
3. Adjectivele relationale, la fel ca substantivele si spre deosebire de adjectivele descriptive, nu au grade de comparatie.
4. Adjectivele relationale nu poseda antonime directe. De aceea ele nu pot fi incorporate in cluster-ele ce caracterizeaza organizarea adjectivelor descriptive.

WordNet mentine un fisier separat de adjective relationale cu pointeri catre substantivele corespunzatoare. Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format PostScript si Acrobat (PDF). Va rugam sa cautati la adresa <http://www.cogsci.princeton.edu/~wn/> sub "Publications".

Cum trateaza WordNet adjectivele relationale? (pusa de doua ori)

WordNet mentine un fisier separat de adjective relationale cu pointeri catre substantivele corespunzatoare.

Circa 1700 synset-uri de adjective relationale, continand peste 3000 de lexeme individuale, sunt incluse in prezent in WordNet. Fiecare synset consta din unul sau mai multe adjective relationale urmate de un pointer catre substantivul corespunzator.

Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format PostScript si Acrobat (PDF). Va rugam sa cautati la adresa <http://www.cogsci.princeton.edu/~wn/> sub "Publications".

Ce semnificatie au numerele care sunt atasate la diverse cuvinte in codificarea cluster-elor de adjective din WordNet? (pusa de doua ori)

Numerele care urmeaza diverselor cuvinte au rolul de a face distinctia intre diferite sensuri secundare sau diferite prioritati de aparitie - spre exemplu, sensul **dried-up1** se refera la o gaura de mina (plina cu apa, dar uscata) si apartine unui synset, in timp ce **dried-up2** se refera la frunze toamna sau la fructe si apartine altui synset. Mai mult, in fiecare dintre aceste cazuri exista informatie inclusa intre paranteze, informatie care sa ajute la distingerea sensului sau care sa indice contexte acceptabile.

Cluster-ele de adjective contin pointeri spre alte cluster-e?

Asa cum se remarca in "Five Papers on WordNet", "pe langa pointerii scrisi cu litere mici din interiorul cluster-elor, multe synset-uri cap contin pointeri spre alte synset-uri inrudite. In cluster-ul AWAKE/ASLEEP, pointerul scris cu majuscule ALERT,& tinteste spre cuvantul cap al cluster-ului ALERT/UNALERT." Acesti pointeri scrisi cu majuscule au sensul de "vezi si" referitor la alte cluster-e inrudite.

Ce imi puteti spune despre cluster-ele de adjective din WordNet care au in capul de cluster cate doua perechi de cuvinte? (pusa de doua ori)

Codificarea restrictionata la interiorul cluster-elor genereaza probleme atunci cand attribute inrudite indeaproape sunt exprimate prin mai multe perechi de antonime. In astfel de cazuri, exact aceeasi multime de synset-uri poate fi pusa in legatura cu doua perechi antonimice diferite, dintre care unele se afla in prezent in cluster-e diferite. (A se lua in considerare **large/small** si **big/little**). Pentru astfel de cazuri a

fost creat un unic cluster, al carui cap contine ambele perechi, evitandu-se in acest mod redundanta inutila. In plus, un anumit synset poate fi codificat cu doi pointeri, unul indreptat spre capul propriului cluster, celalalt spre capul unui cluster din afara.

Este organizarea verbelor din WordNet realizata conform conceptului pe care lingvistii il numesc "domeniu semantic"?

Asa cum se remarca in "Five Papers on WordNet", "verbele sunt impartite in 15 fisiere, in mare parte pe baza criteriilor semantice. Toate aceste fisiere cu exceptia unuia corespund la ceea ce lingvistii au numit domenii semantice: verbe reprezentand functii ale corpului, schimbare, cunoastere, comunicare, competitie, consum, contact, creatie, emotie, miscare, perceptie, posesiune, interactiune sociala si verbe referitoare la vreme. In principiu, toate verbele din aceste fisiere denota evenimente sau actiuni. Un alt fisier contine verbe care se refera la stari, cum ar fi **suffice**, **belong** si **resemble**, care nu au putut fi integrate in celelalte fisiere. Verbele acestui din urma grup nu constituie un domeniu semantic si nu au proprietati semantice in comun in afara faptului ca se refera la stari. Acest fisier, a carui organizare se aseamana cu aceea a adjectivelor in WordNet, este alcatuit din mici cluster-e semantice. Divizarea verbelor in 14 fisiere corespunzand diferitelor domenii semantice, fiecare continand verbe ce desemneaza evenimente si actiuni, dar si un fisier continand verbe diversificate semantic care exprima o stare, reflecta separarea dintre categoriile conceptuale majore EVENIMENT si STARE gasita in analizele lui Jackendoff (1983) si Dowty (1979)."

Care sunt principiile fundamentale care stau la baza relatiilor semantice dintre substantive, adjective si verbe in WordNet? (pusa de doua ori)

Principiul mostenirii lexicale poate fi considerat ca stand la baza relatiilor semantice dintre substantive, in timp ce opozitiile bipolare servesc in organizarea adjectivelor. In mod similar, diferitele relatii care organizeaza verbele pot fi exprimate in termenii unui principiu de baza, si anume implicatia lexicala.

Cum se aseamana relatia de cauzalitate, numita "entailment", dintre verbe cu cea de meronimie dintre substantive, in WordNet? (pusa de doua ori)

Implicatia lexicala dintre verbe se aseamana cu meronimia dintre substantive, dar meronimia se potriveste mai bine substantivelor decat verbelor. Urmatulor exemplu privitor la verbe este oferit in "Five Papers on WordNet":

"Sforaitul sau visatul pot fi o parte a somnului in sensul ca cele doua activitati sunt, macar prtrial, coexistente temporal: timpul pe care il petreci sforaind sau visand este

o parte a timpului pe care il petreci dormind. Si este adevarat ca, atunci cand nu mai dormi, in mod necesar te opresti din sforsait sau visat."

Se spune ca un verb X include un alt verb Y daca exista o perioada de timp in care activitatile desemnate de cele doua verbe au loc simultan, dar nici un interval de timp in care Y intervine iar X nu intervine. Daca exista un interval de timp in care X intervine, dar Y nu intervine, se spune ca X include strict pe Y. O generalizare simpla ar fi urmatoarea: daca X il implica pe Y si daca se verifica o relatie de incluziune temporala intre acestea, atunci vorbitorii vor accepta un enunt de tip parte-intreg care sa lege Y de X.

Ce imi puteti spune despre relatia de hiponimie care se stabileste intre verbe in WordNet? (pusa de doua ori)

Schema de propozitie utilizata pentru testarea hiponimiei in cazul substantivelor, **An X is a Y** (Un X este un Y), nu este adecvata pentru verbe, intrucat pretinde ca X si Y sa fie verbe. Deosebirea semantica dintre doua verbe difera de trasaturile care deosebesc doua substantive in cadrul unei relatii hiponimice.

Numeroasele teorii care disting un "hiponim verbal" de elementul sau supraordonat au fost combinate intr-o unica relatie pe care Fellbaum si Miller (1990) au numit-o **troponimie** (de la grecescul **tropos**, cu sensul maniera, modalitate etc.). Relatia de troponimie dintre doua verbe poate fi exprimata prin formula **To X is to Y in some particular manner** (*A X este a Y intr-un anumit mod*).

Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format PostScript si Acrobat (PDF). Va rugam sa cautati la adresa <http://www.cogsci.princeton.edu/~wn/> sub "Publications".

Este troponimia un caz special de implicatie ("entailment") - cu referire la verbe in WordNet? (pusa de doua ori)

Troponimia este un caz particular de implicatie in sensul ca fiecare troponim X al unui verb mai general Y implica, de asemenea, Y. Pentru ilustrare vom lua in considerare perechea **limp-walk** (a schiopata / a merge), reprezentand exemplul oferit in "Five Papers on WordNet". Autorii comenteaza acest exemplu in felul urmator: "Verbele din acest exemplu se afla intr-o relatie de troponimie: **a schiopata** inseamna, de asemenea, a merge intr-un anumit mod; **a schiopata** este un troponim al lui **a merge**. Verbele se afla si intr-o relatie de implicatie: propozitia **El schiopateaza** implica **El merge**, iar mersul poate fi considerat ca fiind o parte a schiopatului. Spre deosebire de actiunile desemnate de **a sforsai** si **a dormi** sau de **a cumpara** si **a plati**, activitatile la care se refera un troponim si

mai generalul sau supraordonat ocupa intotdeauna aceeași perioadă de timp, în sensul că ceva trebuie neapărat să meargă în fiecare moment în care schiopătează. Troponimia reprezintă, prin urmare, un caz particular de implicatie: perechi care ocupa intotdeauna aceeași perioadă de timp și care sunt legate prin implicatie".

Este adevărat că există mai multe feluri de relații de tip "entailment" cu incluziune temporală în WordNet?

În WordNet sunt discutate două feluri de relații de tip "entailment" cu incluziune temporală. Primul tip este troponimia (a schiopata / a merge), în timp ce implicatia fără troponimie se referă la perechi de verbe (a sfarai / a dormi) legate numai prin implicatie și prin incluziune temporală strictă.

Ce îmi puteți spune despre opoziție și implicatie cu privire la organizarea semantică a verbelor în WordNet?

Așa cum se remarcă în "Five Papers on WordNet", "multe perechi de verbe aflate într-o relație de opoziție au, de asemenea, în comun un verb pe care îl implică. Spre exemplu, atât **hit** (a nimeri), cât și **miss** (a rata) implică **aim** (a ținti), întrucât este nevoie ca ceva să tintească pentru a putea nimeri sau rata ținta". Prin contrast cu alte tipuri de implicatie, "aceste verbe nu sunt asociate prin incluziune temporală. Activitățile desemnate prin **hit** (sau prin **miss**) și prin **aim** au loc într-o ordine secvențială: pentru a nimeri sau a rata ținta, ceva trebuie mai întâi să tintească; țintitul este o precondiție atât pentru a nimeri, cât și pentru a rata". Pentru mai multe informații asupra acestui subiect vă recomandăm să consultați "Five Papers on WordNet", articole disponibile pe web în format PostScript și Acrobat (PDF). Vă rugăm să cautați la adresa <http://www.cogsci.princeton.edu/~wn/> sub "Publications".

Câte feluri de relații de tip "entailment" între verbe au fost luate în considerare în WordNet? (pusă de două ori)

Cele patru tipuri de implicatie între verbe, luate în considerare în WordNet, sunt următoarele:

- implicatie + incluziune temporală + troponimie (a merge / a schiopata);
- implicatie + incluziune temporală - troponimie (a sfarai / a dormi, a cumpara / a plati);
- implicatie - incluziune temporală + presupunere inversă (a reusi / a incerca);
- implicatie - incluziune temporală + cauzalitate (a da / a avea).

Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format PostScript si Acrobat (PDF). Va rugam sa cautati la adresa [http://www.cogsci.princeton.edu/~wn/ sub "Publications"](http://www.cogsci.princeton.edu/~wn/sub%20Publications).

WordNet trateaza toate aspectele sintactice referitoare la verbe?

Asa cum se remarca in "Five Papers on WordNet", pentru a acoperi macar cele mai importante aspecte sintactice legate de verbe, "WordNet include, corespunzator fiecarui synset de verbe, una sau mai multe scheme de propozitie, care specifica caracteristicile de subcategorizare ale verbelor din synset, prin indicarea tipurilor de propozitii in care acestea pot sa intervina. Aceasta informatie permite cautarea rapida printre verbe pentru tipurile de regularitati semantico - sintactice studiate de Levin si de altii".

Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format PostScript si Acrobat (PDF). Va rugam sa cautati la adresa [http://www.cogsci.princeton.edu/~wn/ sub "Publications"](http://www.cogsci.princeton.edu/~wn/sub%20Publications).

Exista fisiere de verbe particulare, specifice la care se face referire in WordNet?

Iata care sunt principalele fisiere cu verbe din WordNet:

- Verbe referitoare la functiile si ingrijirea organismului (sweat, shiver, faint, ache, tire, sleep, freeze);
- Verbe care denota schimbarea (change, alter, vary, modify);
- Verbe de comunicare (beg, order, thank);
- Verbe care denota competitia (fight, race);
- Verbe referitoare la consum (drink, eat);
- Verbe care denota contactul (scrub, wipe, squeeze);
- Verbe ale cunoasterii (reasoning, judging, learning, memorizing);
- Verbe care desemneaza creatia (engrave, sew, bake);
- Verbe de miscare (move, travel);
- Verbe care denota emotii (amuse, encourage);
- Verbe de stare (majoritatea desemneaza starea de a fi si starea de a avea);
- Verbe ale perceptiei (watch, spy, survey);
- Verbe ale posesiunii (hold, own, give, transfer, take, receive);
- Verbe care denota interactiunea sociala (impeach, excommunicate);
- Verbe pentru vreme (rain, thunder, snow).

Pentru mai multe informatii asupra acestui subiect va recomandam sa consultati "Five Papers on WordNet", articole disponibile pe web in format PostScript si Acrobat (PDF). Va rugam sa cautati la adresa <http://www.cogsci.princeton.edu/~wn/> sub "Publications".

III

O SPECIFICAȚIE TEORETICĂ PENTRU UN MODEL MORFOLOGIC AL LIMBII ROMÂNE

GENERAREA FORMELOR FLEXIONARE SUBSTANTIVALE ȘI ADJECTIVALE ÎN LIMBA ROMÂNĂ

Theodor Hristea și Cristian Moroianu

1. Câteva considerații asupra morfologiei românești

Referitor la morfologia limbii române ar fi foarte multe lucruri de spus, întrucât ea este mult mai complicată decât cea englezească și diferă ca origine de aceasta din urmă. Nu trebuie să uităm că engleza este o limbă germanică, iar româna provine din latină, care și ea avea o morfologie extrem de complicată. O altă precizare care se impune este aceea că morfologia engleză este mult mai evoluată și, din această cauză, mai simplă și mai sistematică. Cea românească are, spre exemplu, o flexiune verbală foarte bogată și variată. În acest sens este suficient să precizăm că marele romanist suedez Alf Lombard a consacrat verbului românesc o amplă monografie în două volume, totalizând peste 1200 de pagini. În **Gramatica limbii române** editată de Academia Română și al cărei punct de vedere îl vom urma în prezentul material, se admite că în limba română sunt 4 conjugări, dar există cercetători români sau străini care consideră că noi avem 6, 8, 10, 12 sau chiar 14 conjugări.

Foarte multe și variate sunt, de asemenea, formele pronominale și clasele de adjective (ultimele cu patru terminații, cu trei terminații, numai cu două terminații plus câteva categorii de adjective invariabile). După unii cercetători, există circa 10 declinări la adjectiv. Avem, de asemenea, 4 feluri de articole, dintre care cel mai important este articolul hotărât sau definit. Acesta este, de obicei, postpus (adică situat la sfârșitul cuvântului), dar destul de multe substantive pot primi la genitiv-dativ și un articol proclitic sau antepus.

Substantivul românesc este și el extrem de complicat, din cauză că în limba română există trei genuri: masculin, feminin și neutru. Româna este singura limbă

de origine latină care a păstrat genul neutru, probabil sub influența limbii slave. Având trei genuri, româna are și o flexiune nominală bogată și variată. Mai ales pluralul substantivelor românești se formează cu multe desinențe cărora li se adaugă și alte mărci diferențiatore, dintre care alternanțele fonetice (vocalice și consonantice) sunt cele mai importante. Un exemplu (dintre multe altele) îl constituie substantivul feminin stradă (“street”), care are pluralul străzi. Un altul este substantivul feminin roata (“wheel”), care are pluralul roți. După cum se observă, pluralul se deosebește de singular prin desinența *i* (care se opune lui *ă* de la singular), prin alternanța vocalică *a/ă* și prin cea consonantică *d/z*. În cel de-al doilea caz, alternanța vocalică este *oa/o*, iar cea consonantică este *t/ț* provocată (ca și în primul caz) de prezența lui *i*. Există sute de substantive românești care au două forme de plural, dar, de obicei, una singură este admisă de normele limbii literare. Alteori ambele forme sunt corecte, ceea ce nu se întâmplă în engleză, unde pluralul substantivelor este, cel mai adesea, regulat și se formează prin adăugarea unui *s* la forma de singular.

În continuare vom urma punctul de vedere al **Gramaticii limbii române** editată de Academia Română și vom exemplifica abordarea flexionară a morfologiei în cazul substantivelor și adjectivelor românești.

2. Explicarea paradigmei și a terminologiei folosite

2.1. Paradigma

Cuvintele invariabile sunt ușor de analizat din punct de vedere lexical, întrucât ele au o unică formă de reprezentare. Cele variabile își schimbă forma în diferite situații sintactice, cu alte cuvinte pot fi declinate sau conjugate. Totalitatea formelor flexionare ale unui cuvânt reprezintă **paradigma** acestuia:

P:: {Radical + Flectiv}

În această paradigmă, radicalul poate varia datorită alternanțelor fonetice sau formelor neregulate. Atragem, în mod special, atenția asupra folosirii termenului de *radical* și nu a celui de *rădăcină*, o diferențiere asupra căreia ne vom opri mai pe larg în cele ce urmează. Prin **radical** (engl. “stem”, “theme” sau “thema”) înțelegem rădăcina cuvântului (engl. “root”) la care se adaugă eventuale afixe/infixe. Este vorba despre baza flexionară a unui cuvânt, căreia îi pot fi adăugate alte elemente: vocale de legătură, desinențe etc. Evident, în multe cazuri, radicalul poate fi identic cu rădăcina unui cuvânt. Spre deosebire de radical, a cărui existență este obligatorie, flectivul poate fi nul. Flectivul variază în concordanță cu categoriile morfologice ale părții de vorbire. Spre exemplu, în cazul substantivelor, această variație este determinată de număr, caz, gen și articulare. În cazul verbelor ea corespunde modului, timpului, numărului și persoanei. În cazul aceluiași părți de vorbire, flectivul corespunzând diferitelor categorii morfologice poate avea forme diferite. Acesta este criteriul conform căruia cuvintele diferite, dar care se încadrează în aceeași parte de vorbire, sunt grupate în clase flexionare. Flectivele care caracterizează o anumită clasă flexionară sunt aceleași pentru toate cuvintele aparținând acelei clase. În cazul generării formelor flexionare substantive și adjective românești, s-a considerat (a se vedea 3.) că flectivul fie este nul, fie poate fi alcătuit din vocală de legătură și articol hotărât (atunci când este cazul) sau din desinență și articol hotărât (atunci când este cazul).

2.2. Raportul dintre rădăcină și radical

Așa cum se arată în [2], în majoritatea lucrărilor de specialitate (românești și străine), termenul de rădăcină este sinonim cu cel de radical, deși nu este normal să se folosească doi termeni diferiți pentru exact aceeași realitate lingvistică. După Valeria Guțu-Romalo (vezi *Morfologie structurală a limbii române*, București, 1968, p.39 și urm.), rădăcina poate să coincidă cu radicalul ori poate fi inclusă în acesta din urmă atunci când e vorba de cuvinte formate prin derivare. Astfel, la nivelul limbii române, un segment fonic cum este **cânt** – (din *cânt-a*) trebuie

considerat, în același timp, rădăcină și radical, însă în *descânt-a* radicalul este **descânt-**, ceea ce înseamnă că el coincide cu așa-zisa “temă lexicală”. Tot așa, în *călători* rădăcina este **căl-** (din *cal-e*), pe când radicalul e o grupare de două morfeme (**căl-ător**), deci o unitate divizibilă, în a cărei componență intră și rădăcina privită exclusiv ca “morfem independent” sau “unitate morfemică indivizibilă”. Precum vedem, radicalul poate să conțină în plus anumite afixe derivate, pe când rădăcina este întotdeauna o unitate minimală indivizibilă. Indiferent dacă el coincide cu rădăcina (ca în *bat* – *e*, *cânt* – *a* etc.) sau nu se identifică cu aceasta (ca în *răzbat-e*, *încânt-a* și altele), radicalul apare ca element constant în toate formele flexionare ale unui cuvânt, fie el derivat sau nederivat. Acceptând această distincție, care se întâlnește și la unii lingviști străini și pe care o considerăm binevenită, cei doi termeni (adică *rădăcină* și *radical*) pot fi folosiți, în continuare, precis specializați din punct de vedere semantic. Pentru unele detalii și mai ales pentru raportul care există între *radical* și *flectiv*, vezi paragraful **Structura morfematică a cuvintelor** din [2].

3. Generarea formelor flexionare substantive si adjectivale

3.1. SUBSTANTIVE

În cazul substantivelor românești există trei genuri (masculin, feminin, neutru) și sunt cunoscute două tipuri principale de flexiune: cu articol hotărât și cu articol nehotărât. Corespunzător fiecăreia dintre regulile următoare sunt prezentate atât flexiunea cu articol hotărât, cât și cea cu articol nehotărât. Vor fi utilizate următoarele abrevieri: “N” pentru cazul nominativ, “AC” pentru acuzativ, “G” pentru genitiv, “D” pentru dativ, “sg.” pentru singular, “pl.” pentru plural, “Rad.” pentru radical, “Voc.” pentru vocală de legătură, “Des.” pentru desinență și “Art .” pentru articol.

3.1.a. SUBSTANTIVE MASCULINE

Regula nr. 1

Articol hotărât

sg.

N + AC: Rad. + Voc. - **u** + Art. - **l**

G + D: Rad. + Voc. - **u** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **i** + Art. - **i**

G + D: Rad. + Des. - **i** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad.

G + D: Rad.

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

Regula nr. 2

Articol hotărât

sg.

N + AC: Rad. + Des. - **e** + Art. - **le**

G + D: Rad. + Des. - **e** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **i** + Art. - **i**

G + D: Rad. + Des. - **i** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

3.1.b. SUBSTANTIVE NEUTRE

Regula nr.3

Articol hotărât

sg.

N + AC: Rad. + Voc. - **u** + Art. - **l**

G + D: Rad. + Voc. - **u** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **uri** + Art. - **le**

G + D: Rad. + Des. - **uri** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad.

G + D: Rad.

pl.

N + AC: Rad. + Des. - **uri**

G + D: Rad. + Des. - **uri**

Regula nr. 4

Articol hotărât

sg.

N + AC: Rad. + Voc. - **u** + Art. - **l**

G + D: Rad. + Voc. - **u** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **e** + Art. - **le**

G + D: Rad. + Des. - **e** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad.

G + D: Rad.

pl.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

EXCEPȚIE:

- altemanță vocalică în radical: N,G,D,AC pl. - **o/oa**

Regula nr. 5

Articol hotărât

sg.

N + AC: Rad. + Voc. - **u** + Art. - **l**

G + D: Rad. + Voc. - **u** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **i** + Art. - **le**

G + D: Rad. + Des. - **i** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad. + Des. - **u**

G + D: Rad. + Des. - **u**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

Regula nr. 6

Articol hotărât

sg.

N + AC: Rad. + Des. - **e** + Art. - **le**

G + D: Rad. + Des. - **e** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **e** + Art. - **le**

G + D: Rad. + Des. - **e** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

pl.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

3.1.c. SUBSTANTIVE FEMININE

Regula nr. 7

Articol hotărât

sg.

N + AC: Rad. + Des. - **e** + Art. - **a**

G + D: Rad. + Des. - **i** + Art. - **i**

pl.

N + AC: Rad. + Des. - **i** + Art. - **le**

G + D: Rad. + Des. - **i** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **i**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

EXCEPTIE:

- alternanță vocalică în radical: G,D sg. / N,G,D,AC pl. - **a/ă**

Regula nr.8

Articol hotărât

sg.

N + AC: Rad. + Art. - **a**

G + D: Rad. + Des. - **e** + Art. - **i**

pl.

N + AC: Rad. + Des. - **i** + Art. - **le**

G + D: Rad. + Des. - **i** + Art. - **lor**

Articol nehotărât

sg.

N + AC: Rad. + Des. - **e**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - i

G + D: Rad. + Des. - i

Regula nr. 9

Articol hotărât

sg.

N + AC: Rad. + Art. - a

G + D: Rad. + Des. - e + Art. - i

pl.

N + AC: Rad. + Des. - e + Art. - le

G + D: Rad. + Des. - e + Art. - lor

Articol nehotărât

sg.

N + AC: Rad. + Des. - ă

G + D: Rad. + Des. - e

pl.

N + AC: Rad. + Des. - e

G + D: Rad. + Des. - e

EXCEPȚIE:

- alternanță vocalică în radical: G,D sg. / N,G,D,AC pl. - ea/e

Regula nr. 10

Articol hotărât

sg.

N + AC: Rad. + Art. - a

G + D: Rad. + Des. - i + Art. - i

pl.

N + AC: Rad. + Des. - i + Art. - le

G + D: Rad. + Des. - i + Art. - lor

Articol nehotărât

sg.

N + AC: Rad. + Des. - ă

G + D: Rad. + Des. - i

pl.

N + AC: Rad. + Des. - i

G + D: Rad. + Des. - i

3.2. ADJECTIVE

Regulile de flexiune referitoare la adjective, în limba română, depind, în egală măsură, de tipul de articol care intervine (hotărât sau nehotărât), dar și de topica adjectivului în raport cu substantivul pe care îl determină (adjectiv antepus sau postpus). Implementarea noastră face distincția între următoarele tipuri de reguli (în cadrul cărora abrevierile sunt aceleași dinainte):

Regula nr. 1

Această regulă se referă la **adjective cu patru terminații și care nu prezintă alternanțe fonetice în radical**. În cele ce urmează, regula va fi descrisă ținându-se cont de tipul articolului, de topica și de genul adjectivului. Vom mai nota faptul că, în limba română, în cazul tuturor regulilor de flexiune a adjectivului avem

Neutru SG. = Masculin SG. și Neutru PL. = Feminin PL.

acesta fiind și motivul pentru care genul neutru nu va fi luat în considerare în mod separat. Regula nr. 1 poate fi descrisă, conform criteriilor de mai sus, după cum urmează:

1.1. Articol nehotărât (cu adjectiv postpus sau antepus substantivului) + articol hotărât (cu adjectiv postpus):

Masculin

sg.

N + AC: Rad.

pl.

N + AC: Rad. + Des. - i

G + D: Rad.

G + D: Rad. + Des. - i

Feminin

sg.

pl.

N + AC: Rad. + Des. - ă

N + AC: Rad. + Des. - e

G + D: Rad. + Des. - e

G + D: Rad. + Des. - e

1.2. Articol hotărât cu adjectiv antepus:

Masculin

sg.

pl.

N + AC: Rad. + Voc. - u + Art. - l

N + AC: Rad. + Des. - i + Art. - i

G + D: Rad. + Voc. - u + Art. - lui

G + D: Rad. + Des. - i + Art. - lor

Feminin

sg.

pl.

N + AC: Rad. + Art. - a

N + AC: Rad. + Des. - e + Art. - le

G + D: Rad. + Des. - e + Art. - i

G + D: Rad. + Des. - e + Art. - lor

EXCEPȚII:

- în cazul adjectivelor al căror radical se termină în consoana t, același radical se termină în ț la **masculin plural** - alternanță fonetică (consonantică) în cadrul radicalului la masculin plural (atât la 1.1, cât și la 1.2): t/ț;
- în cazul adjectivelor al căror radical se termină în consoana s, același radical se termină în ș la **masculin plural** - alternanță fonetică (consonantică) în cadrul radicalului la masculin plural: s/ș;

- în cazul adjectivelor al căror radical se termină în consoana **d**, același radical se termină în **z** la **masculin plural** - alternanță fonetică (consonantică) în cadrul radicalului la masculin plural: **d/z**;
- în cazul adjectivelor care conțin vocala **o** în ultima silabă, intervine o alternanță fonetică (vocalică) în cadrul radicalului, la **feminin singular și plural**: **o/oa**;
- în cazul adjectivelor care conțin vocala **ă** în ultima silabă, intervine o alternanță fonetică (vocalică) în cadrul radicalului, la **masculin și feminin plural**: **ă/e**.

Regula nr. 2

Această regulă se referă la **adjective cu patru terminații al căror radical se termină într-o vocală și care prezintă o desinență diferită la masculin singular**.

Regula poate fi descrisă, conform aceluiași criterii, după cum urmează:

2.1. Articol nehotărât (cu adjectiv postpus sau antepus substantivului) + articol hotărât (cu adjectiv postpus):

Masculin

sg.

N + AC: Rad. + Des. - **u**

G + D: Rad. + Des. - **u**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

Feminin

sg.

N + AC: Rad. + Des. - **ă**

G + D: Rad. + Des. - **e**

pl.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

2.2. Articol hotărât cu adjectiv antepus:

Masculin

sg.

N + AC: Rad. + Des. - **u** + Art. - **l**

G + D: Rad. + Des. - **u** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **i** + Art. - **i**

G + D: Rad. + Des. - **i** + Art. - **lor**

Feminin

sg.

N + AC: Rad. + Art. - **a**

G + D: Rad. + Des. - **e** + Art. - **i**

pl.

N + AC: Rad. + Des. - **e** + Art. - **le**

G + D: Rad. + Des. - **e** + Art. - **lor**

Regula nr. 3

Această regulă se referă la **adjective cu trei terminații al căror radical se termină într-o consoană**; întrucât aceste adjective conțin vocala “o” în ultima silabă, ele prezintă alternanța fonetică **o/oa** la feminin singular și plural. Regula poate fi descrisă, conform aceluiași criterii, după cum urmează:

3.1. Articol nehotărât (cu adjectiv postpus sau antepus substantivului) + articol hotărât (cu adjectiv postpus):

Masculin

sg.

N + AC: Rad.

G + D: Rad.

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

Feminin

sg.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

pl.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

3.2. Articol hotărât cu adjectiv antepus:

Masculin

sg.

N + AC: Rad. + Voc. - **u** + Art. - **l**

G + D: Rad. + Voc. - **u** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **i** + Art. - **i**

G + D: Rad. + Des. - **i** + Art. - **lor**

Feminin

sg.

N + AC: Rad. + Des. - **e** + Art. - **a**

G + D: Rad. + Des. - **e** + Art. - **i**

pl.

N + AC: Rad. + Des. - **e** + Art. - **le**

G + D: Rad. + Des. - **e** + Art. - **lor**

Regula nr. 4

Această regulă se referă, de asemenea, la **adjective având trei terminații**. În cadrul acestei clase **radicalul se termină însă într-o vocală**, iar distribuția desinențelor este diferită. Regula poate fi descrisă, conform aceluiași criterii, după cum urmează:

4.1. Articol nehotărât (cu adjectiv postpus sau antepus substantivului) + articol hotărât (cu adjectiv postpus):

Masculin

sg.

N + AC: Rad. + Des. - **u**

G + D: Rad. + Des. - **u**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

Feminin

sg.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **i**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

4.2. Articol hotărât cu adjectiv antepus:

Masculin

sg.

N + AC: Rad. + Des. - **u** + Art. - **l**

G + D: Rad. + Des. - **u** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **i** + Art. - **i**

G + D: Rad. + Des. - **i** + Art. - **lor**

Feminin

sg.

N + AC: Rad. + Art. - **a**

G + D: Rad. + Des. - **e** + Art. - **i**

pl.

N + AC: Rad. + Des. - **i** + Art. - **le**

G + D: Rad. + Des. - **i** + Art. - **lor**

Regula nr. 5

Această regulă se referă, de asemenea, la **adjective având trei terminații**. **Radicalul se termină într-o consoană**, dar distribuția desinențelor este diferită de cea de la regula nr. 3. Regula poate fi descrisă, conform acelorași criterii, după cum urmează:

5.1. Articol nehotărât (cu adjectiv postpus sau antepus substantivului) + articol hotărât (cu adjectiv postpus):

Masculin

sg.

N + AC: Rad.

G + D: Rad.

pl.

N + AC: Rad. + Des. - i

G + D: Rad. + Des. - i

Feminin

sg.

N + AC: Rad. + Des. - ă

G + D: Rad. + Des. - i

pl.

N + AC: Rad. + Des. - i

G + D: Rad. + Des. - i

5.2. Articol hotărât cu adjectiv antepus:

Masculin

sg.

N + AC: Rad. + Voc. - u + Art. - l

G + D: Rad. + Voc. - u + Art. - lui

pl.

N + AC: Rad. + Des. - i + Art. - i

G + D: Rad. + Des. - i + Art. - lor

Feminin

sg.

N + AC: Rad. + Art. - **a**

G + D: Rad. + Des. - **i** + Art. - **i**

pl.

N + AC: Rad. + Des. - **i** + Art. - **le**

G + D: Rad. + Des. - **i** + Art. - **lor**

EXCEPȚIE:

- alternanță fonetică în radical la feminin sg., N + AC: **e/ea**

Regula nr. 6

Această regulă se referă la **adjective având două terminații și un radical care se termină într-o consoană**. Regula poate fi descrisă, conform aceluiași criterii, după cum urmează:

6.1. Articol nehotărât (cu adjectiv postpus sau antepus substantivului) + articol hotărât (cu adjectiv postpus):

Masculin

sg.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **e**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

Feminin

sg.

N + AC: Rad. + Des. - **e**

G + D: Rad. + Des. - **i**

pl.

N + AC: Rad. + Des. - **i**

G + D: Rad. + Des. - **i**

6.2. Articol hotărât cu adjectiv antepus :

Masculin

sg.

N + AC: Rad. + Des. - **e** + Art. - **le**

G + D: Rad. + Des. - **e** + Art. - **lui**

pl.

N + AC: Rad. + Des. - **i** + Art. - **i**

G + D: Rad. + Des. - **i** + Art. - **lor**

Feminin

sg.

N + AC: Rad. + Des. - **e** + Art. - **a**

G + D: Rad. + Des. - **i** + Art. - **i**

pl.

N + AC: Rad. + Term - **i** + Art. - **le**

G + D: Rad. + Des. - **i** + Art. - **lor**

Regula nr. 7

Această regulă se referă la adjective având o singură terminație - **adjective invariabile**. În toate situațiile, ele constau numai dintr-un radical, care se termină într-o vocală.

Bibliografie:

1. Gramatica limbii române (vol. I). Ediția a doua revăzută și adăugită. București, Editura Academiei, 1966, pp.41-134.
2. Theodor Hristea (coordonator), Sinteze de limba română. Ediția a treia revăzută și din nou îmbogățită. București, Editura "Albatros", 1984, pp. 67-70; 203-224.
3. Luciana Peev, Lidia Bibolar, Jodal Endre, "A Formalization Model of the Romanian Morphology", în Recent Advances in Romanian Language Technology (editori Dan Tufiş și Poul Andersen). Editura Academiei Române, București, 1997.

PREMISE ALE UNUI DICȚIONAR MORFOLOGIC ELECTRONIC AL LIMBII ROMÂNE

Emil Ionescu

1. Scop

Obiectivul prezentului demers este construcția unui dicționar electronic de forme flexionare adnotate al limbii române¹. În forma sa de acum (care este o formă incipientă) dicționarul se găsește în documentul *Romdict.txt* și este exprimat în formatul DELAF (INTEX, Zilberstein 1993). Dicționarul este asociat cu un alt document care conține descrierile etichetelor utilizate în adnotare (*Definitions.txt*). Amîndouă documentele servesc drept resurse pentru analizorul construit de partenerii bulgari din acest proiect. Analizorul este disponibil la adresa <http://www.larflast.bas.bg/balric/tag/default.htm>.

Dicționarul se bazează pe utilizarea trăsăturilor morfologice și este un lexicon cu forme flexionare nedescompuse ("a full-forms lexicon"). Pentru că nu se ocupă cu analiza morfematică, dicționarul nu conține rădăcini și afixe gramaticale. Pe de altă parte, nu sînt luate în considerație nici afixele derivative. Principala preocupare a fost aceea de a capta proprietățile formelor lexicale în termenii trăsăturilor specifice morfologiei limbii române. Aceasta este o direcție de cercetare reprezentativă pentru momentul actual în Tehnologia Limbajului Uman. Este vorba despre o direcție care privilegiază corelația dintre *text* și elementele lui componente ultime, *cuvintele*.

2. Structura generală a dicționarului

O intrare a acestui dicționar are structura următoare:

- Forma flexionară
- Lema
- Caracteristici (care pot fi caracteristici ale formei flexionare sau ale lemei)

¹ Pentru instrumente și resurse privind morfologia limbii române create pînă în prezent, a se vedea Tufis, L. Diaconu, Barbu and C. Diaconu (1996), Tufis, L. Diaconu, C. Diaconu și Barbu (1996), Peev, Bibolar și Endre (1996), Curteanu, Holban, I. Pavalo, C. Pavalo, Negulescu și Todirascu (1996), Vuscan (1996) – toate contribuțiile menționate fiind prezentate în volumul Dan Tufis (ed.) "Limbaj si Tehnologie", Editura Academiei Române, Bucuresti, 1996.

De exemplu:

Formă flexionară: agresivă

Lemă: agresiv

Caracteristici ale lemei: A (adjectiv) + GR (cu grade de comparație)

Caracteristici ale formei flexionare: ufsr (nedefinit, feminin, singular, nominativ sau acusativ)

Dicționarul conține numai forme sintetice. Formele flexionare analitice sînt considerate colocații. Formele sintetice sînt la rîndul lor de două feluri: simple și compuse. De pildă, agresivă este o formă flexionară simplă, în timp ce nici un este compusă. Formele compuse sînt notate cu linioară la piciorul literei. În forma sa actuală, dicționarul cuprinde 6768 de forme, care acoperă toate părțile de vorbire din limba română.

3. Etapele cercetării

Principalii pași care au condus la rezultatele mai sus-menționate au fost următorii:

- Stabilirea corpusului pe care se bazează dicționarul
- Identificarea formelor lexicale care aparțin corpusului
- Construcția paradigmelor morfologice (acolo unde este nevoie)
- Stabilirea trăsăturilor morfologice implicate în caracterizarea lemelor sau a formelor flexionare

3.1. Corpusul

Corpusul este reprezentat de un grup de 13 articole extrase dintr-unul din cele mai reprezentative cotidiane naționale „Evenimentul Zilei”. Corpusul este caracteristic pentru româna contemporană standard.

3.2. Identificarea formelor lexicale

Articolele de ziar au furnizat un număr de 1478 de forme lexicale, care au fost izolate cu ajutorul unui segmentator lexical („tokenizer”), construit la RACAI în cadrul proiectului MULTEX (www.racai.ro). Un eșantion de text segmentat lexical este cel de mai jos:

#el #

#a#

#afirmat#

#că#

#singura#

#soluție#
 #pentru_ca #
 #populația#
 #să #
 #-și#
 #poată #
 #achita#
 #facturile#
 #este#
 #creșterea#
 #reală #
 #a#
 #salariilor#

Formele identificate prin segmentare aparțin la 523 de leme care simbolizează atât părți de vorbire flexibile cât și neflexibile.

3.3. Paradigme flexionare

Pentru fiecare leamă aparținând unei părți de vorbire flexibile, s-a construit paradigma corespunzătoare. Operația s-a făcut manual. Rezultatul a fost producerea a 6768 de forme flexionare, care reprezintă forma actuală a dicționarului.

3.4. Trăsături morfologice

3.4.1 Stabilirea setului de trăsături morfologice pentru limba română a reprezentat componenta lingvistică a cercetării. Am urmat în această privință modelul dicționarului bulgar . Aceasta înseamnă că o trăsătură dată a fost considerată relevantă din punct de vedere morfologic, dacă ea a fost găsită importantă pentru producerea și/sau distingerea membrilor unei paradigme. De exemplu, trăsătura A(uxiliar) este în română o trăsătură sintactică a verbelor. Cu toate acestea, ea poate fi considerată în același timp o trăsătură morfologică, deoarece paradigmele verbului auxiliar **a avea** și **a vrea** sunt distincte de corespondentele lor predicative. Într-un mod asemănător, în cazul adjectivelor, genul – o trăsătură semantică – trebuie și el considerat o trăsătură morfologică – întocmai ca în bulgară, dar deosebit de engleză – deoarece el ajută la distingerea diverșilor membri ai aceleași paradigme.

3.4.2. Cu aceste criterii la dispoziție, au fost identificate douăsprezece părți principale de vorbire, și anume, numele, verbul, adjectivul, determinatorii, pronumele, numeralul, articolul, adverbul, prepoziția, interjecția, particulele și abrevierea.

Ultimele două părți de vorbire nu se întâlnesc în mod curent în descrierile morfologice ale limbii române, și de aceea sunt necesare câteva explicații în acest sens. Categoria particulelor conține câteva cuvinte cu distribuție și comportament speciale. Este vorba despre ‘adverbul’ de negație **nu**, de conjuncția **să**, și de așa-numitul morfem **a**, care marchează infinitivul.

În ceea ce privește abrevierile (de exemplu, **tel** de la **telefon**), opțiunea de a le adopta în calitate de categorii distinctă se justifică prin faptul că în general o abreviere diferă în mod semnificativ de partea de vorbire completă care îi este analogă. De exemplu, **tel** nu prezintă flexiune de caz și nici nu poate fi utilizat la plural. Oricum, dat fiind caracterul marginal al abrevierilor – în mod special din punct de vedere cantitativ – surprinzătoarea lor prezență în inventarul părților de vorbire ale limbii române nu pune probleme speciale.

3.4.3. Se poate întâmpla ca aceeași trăsătură să fie considerată drept o caracteristică a lemei în raport cu o anumită parte de vorbire, și o caracteristică a formei flexionare în raport cu alta. De pildă, genul caracterizează lema substantivelor dar forma flexionară a adjectivelor.

3.4.4. Există o ușoară diferență între felul în care dicționarul bulgar și cel românesc tratează formele flexionare omonime. Diferența poate fi observată în tratamentul următorului exemplu ipotetic. Pentru o formă flexionară precum **fly**, în dicționarul bulgar, analiza este timpul prezent, persoana 1 sau 2 singular, sau persoana 1, 2, sau 3 plural. În dicționarul românesc în schimb, cuvântul este analizat **fly**, timpul prezent, persoana 1 sau 2 singular; **fly** timpul prezent, persoana 1, 2 sau 3 plural. Prin urmare, forma flexionară este înregistrată de două ori (dacă, firește, omonimia este ilustrată de două forme.)

3.4.5. Trăsăturile morfologice ale limbii române reținute în prezentul dicționar sînt următoarele

Substantiv

Trăsături ale lemei

- Substantiv
- Comun sau propriu
- Masculin sau feminin sau neutru

Trăsături ale formei flexionare

- Singular sau plural
- Definit sau nedefinit
- Nominativ sau acuzativ; genitiv sau dativ

Verb

Trăsături ale lemei

- Verb
- Auxiliar (trăsătură prin exceptare)

Trăsături ale formei flexionare

- Indicativ sau conjunctiv sau imperativ sau infinitiv sau gerunziu sau participiu
- Prezent sau imperfect sau perfectul simplu sau mai mult ca perfect
- Persoana 1 sau 2 sau 3
- Singular sau plural

Adjectivul

Trăsături ale lemei

- Adjectiv
- Cu sau fără grad de comparație

Trăsături ale formei flexionare

- Singular sau plural
- Definit sau nedefinit
- Nominativ sau acuzativ sau genitiv sau dativ

Pronume

Trăsături ale lemei

- Pronume
- Personal sau reflexiv sau demonstrativ sau negativ sau nedefinit sau interogativ-relativ

Trăsături ale formei flexionare

Pentru pronumele personale

- Formă accentuată sau neaccentuată
- Persoana 1 sau 2 sau 3
- Singular sau plural
- Nominativ sau acuzativ sau genitiv sau dativ

- Masculin sau feminin

Pentru pronumele reflexive

- Formă accentuată sau neaccentuată
- Persoana 1 sau 2 sau 3
- Singular sau plural
- Nominativ sau acuzativ sau dativ

Pentru pronumele demonstrative

- Singular sau plural
- Nominativ sau acuzativ sau dativ

Pentru pronumele negative

- Singular sau plural
- Nominativ sau acuzativ sau dativ

Pentru pronumele interogativ-relative

- Nominativ sau acuzativ sau genitiv sau dativ

Pentru pronumele nehotărâte

- Nominativ sau acuzativ sau genitiv sau dativ

Determinatori

Trăsături ale lemei

- Determinator
- Demonstrativ sau posesiv sau negativ sau nehotărât sau interogativ-relativ sau de întărire

Trăsături ale formei flexionare

Pentru determinatori demonstrativi

- Singular sau plural
- Nominativ sau acuzativ sau genitiv sau dativ

Pentru determinatori negativi

- Singular sau plural
- Nominativ sau acuzativ sau genitiv sau dativ

Pentru determinatori nehotărâți

- Singular sau plural
- Nominativ sau acuzativ sau genitiv sau dativ

Pentru determinatori interogativ-relativi

- Singular sau plural
- Nominativ sau acuzativ sau genitiv sau dativ

Pentru determinatori posesivi

- Persoana 1, 2 sau 3
- Singular sau plural

Pentru determinatori de întărire

- Persoana 1 sau 2 sau 3
- Singular sau plural
- Nominativ sau acuzativ sau genitiv sau dativ

Articol

Trăsături ale lemei

- Articol
- Demonstrativ sau posesiv sau nehotărât

Trăsături ale formei flexionare

- Masculin sau feminin
- Singular sau plural
- Nominativ sau acuzativ sau genitiv sau dativ

Numeral

Trăsături ale lemei

- Numeral
- Cardinal sau ordinal

Trăsături ale formei flexionare

- Masculin sau feminin
- Singular sau plural

Adverb

Trăsături ale lemei

- Adverb
- General sau verbal sau interogativ-relativ

Conjuncție

Trăsături ale lemei

- Conjuncție
- Coordonatoare sau subordonatoare

Abreviere

Trăsături ale lemei

- Abreviere

Buletinul RORIC-LING

Iunile 13 - 18

Au fost puse **65** de intrebari de catre utilizatori provenind in special din unitatile de invatamant romanesti, dar nu numai. Firmele de software atat din Romania, cat si din strainatate sunt, de asemenea, reprezentate. Unele dintre intrebari au fost puse de mai multe ori, asa cum va fi specificat in buletin. Exista doua mari categorii de intrebari, corespunzand tematicii partii romanesti a proiectului BALRIC-LING, dupa cum urmeaza:

- intrebari generale privitoare la cele doua abordari ale morfologiei luate in discutie;
- intrebari referitoare la dictionarul morfologic si implementarea corespunzatoare acestuia.

In interiorul buletinului, intrebarile au fost grupate in conformitate cu tematica la care se refera (si nu in ordinea subscrierii, adica in ordine cronologica). Toate informatiile referitoare la numele si datele personale ale utilizatorilor exista in fisierele RORIC-LING, dar au fost sterse din buletin, pentru a facilita citirea si folosirea acestui material, precum si cautarea in cadrul lui dupa tematica.

Cateva statistici:

Intrebari: 65

Tara: **Romania Alta***
Intrebari: 41 24

Limba Materna: **Romana Alta**
Intrebari: 52 13

Domeniu de Activitate: **Educatie Cercetare Industrie Software Altul**
Intrebari: 29 17 12 7

*** AUSTRALIA, CANADA, GERMANY, ITALY, UNITED KINGDOM, UNITED STATES**

Intrebări generale privitoare la cele doua abordari ale morfologiei luate in discutie

Ce este, in mod cat mai exact, un dictionar morfologic? (pusa de doua ori)

Un dictionar morfologic este un rezumat reprezentativ al tuturor formelor lexicale de baza dintr-o anumita limba, insotite de caracteristicile lor gramaticale. Aceste trasaturi determina generarea tuturor formelor lexicale care sunt derivate din cea de baza si furnizeaza informatia de baza pentru rezultatele analizei textului. Dictionarele morfologice sunt printre primele aplicatii din domeniul procesarii limbajului natural si reprezinta un instrument esential in colectarea si organizarea datelor lingvistice.

Dictionarul morfologic este o baza de date care furnizeaza o gama larga de informatii referitoare la caracteristicile morfologice si la formele unui cuvânt dat. El permite, de asemenea, regasirea rapida a informatiilor gramaticale care provin simultan de la paradigme diferite. Principalul tel al unui dictionar morfologic este sa identifice relatiile dintre o forma lexicala concreta si invarianta ei (lema). Scopul dictionarului morfologic este, prin urmare, sa identifice forma lexicala si caracteristicile acesteia si sa o clasifice in raport cu lema sa.

Pentru mai multe informatii asupra dictionarelor morfologice, va invitam sa consultati pagina bulgara a proiectului BALRIC-LING, la adresa

http://www.larflast.bas.bg/balric/index/index_eng.htm

Care este diferenta dintre radacina si radical? (pusa de doua ori)

Prin radical (engl. "stem", "theme" sau "thema") intelegem radacina cuvantului (engl. "root") la care se adauga eventuale afixe/infixe. Este vorba despre baza flexionara a unui cuvânt, careia ii pot fi adaugate alte elemente: vocale de legatura, desinente etc. Evident, in multe cazuri, radicalul poate fi identic cu radacina unui cuvânt.

In majoritatea lucrarilor de specialitate (romanesti si straine), termenul de radacina este sinonim cu cel de radical, desi nu este normal sa se foloseasca doi termeni diferiti pentru exact aceeasi realitate lingvistica. Dupa Valeria Gutu-Romalo (vezi Morfologie structurala a limbii romane, Bucuresti, 1968, p.39 si urm.), radacina poate sa coincidă cu radicalul ori poate fi inclusa in acesta din urma atunci cand e vorba de cuvinte formate prin derivare. Astfel, la nivelul limbii romane, un segment fonic cum este cant - (din cant-a) trebuie considerat, in acelasi timp, radacina si radical, inasa in descant-a radicalul este descant-, ceea ce inseamna ca el coincide cu asa-zisa "tema lexicala". Tot asa, in calatori radacina este cal- (din cal-

e), pe cand radicalul e o grupare de doua morfeme (cal-ator), deci o unitate divizibila, in a carei componenta intra si radacina privita exclusiv ca "morfem independent" sau "unitate morfemica indivizibila". Precum vedem, radicalul poate sa contina in plus anumite afixe derivative, pe cand radacina este intotdeauna o unitate minimala indivizibila. Indiferent daca el coincide cu radacina (ca in bat-e, cant-a etc.) sau nu se identifica cu aceasta (ca in razbat-e, incant-a si altele), radicalul apare ca element constant in toate formele flexionare ale unui cuvânt, fie el derivat sau nederivat. Acceptand aceasta distinctie, care se intalneste si la unii lingvisti straini si pe care o consideram binevenita, cei doi termeni (adica radacina si radical) pot fi folositi precis specializati din punct de vedere semantic.

Care sunt avantajele utilizarii dictionarului morfologic (prin comparatie cu abordarea flexionara)? (pusa de 3 ori)

Principalele avantaje sunt:

1. Se evita o data pentru totdeauna discutia "dureroasa" purtata la nivel morfematic si atentia se concentreaza direct asupra cuvintelor, care au rol de "caramida" (dar toate formele lexicale sunt luate in considerare in mod separat). Flexiunea se refera la particularitatile de formare a cuvintelor intr-o limba data. Intrucat astazi tehnologia limbajului este preocupata in special de analiza textului in cazul majoritatii limbilor europene, este necesara punerea in evidenta a legaturii cuvânt - text, o sarcina careia dictionarul morfologic ii este extrem de util.
2. Suprapunerea oricarui text peste un asemenea lexicon permite discutarea problemelor de POS-dezambiguizare, despre care se considera astazi ca reprezinta adevaratele probleme ale analizei textului (la nivelul cuvântului).

Care este criteriul esential pe baza caruia o anumita trasatura va fi inclusa in dictionarul morfologic? (pusa de 3 ori)

Principalul criteriu de includere a unei trasaturi poate fi exprimat prin intermediul urmatoarei intrebari: "Este acea trasatura importanta pentru producerea si distingerea membrilor paradigmei?"

Intrebari referitoare la dictionarul morfologic si implementarea corespunzatoare acestuia

Poate fi folosit un dictionar morfologic la construirea unui spell checker pentru limba romana? (pusa de doua ori)

Nu cred ca in mod direct, dar avand in vedere ca intr-un astfel de dictionar sunt prezente in mod explicit toate formele flexionare ale unui cuvânt, dictionarul (unul real, nu o mostra) ar putea fi folosit la alcătuirea unei liste de cuvinte, care mai apoi ar putea fi folosită la construcția unui spell checker. O astfel de soluție nu ar exploata decât o mică parte din informația prezentă într-un dictionar morfologic.

Care este diferența dintre un dictionar al formelor flexionare complete și un dictionar de morfologie flexionară (derivatională)? (pusa de două ori)

Diferența principală este că într-un dictionar al formelor flexionare complete nu există reprezentarea structurii cuvântului.

De ce ați ales doar articole de ziar drept surse de corpus? (pusa de două ori)

Articolele de ziar sunt reprezentative pentru starea unei limbi la un moment dat, și de altfel în munca de alcătuire a corpusurilor este o practică uzuală să se folosească astfel de esanțioane.

Nu este și tranzitivitatea o trasătură interesantă din punct de vedere morfologic? Care este motivul pentru care nu a fost inclusă printre trasăturile verbului? (pusa de două ori)

Motivul a fost că tranzitivitatea unui verb nu este marcată în limba română morfologic.

Veti extinde dictionarul? (pusa de doua ori)

Bineînțeles, dorim să facem asta, dar totul depinde mai departe de oportunitățile legate de un nou proiect.

Mai exista si alte contributii on-line la realizarea de resurse morfologice pentru limba romana? (pusa de doua ori)

Din cate stim noi, contributii on-line nu exista, dar exista proiecte in desfasurare la Institutul de Inteligenta Artificiala din Bucuresti si la laboratorul de lingvistica computationala de la Cluj.

Diateza pasiva este o categorie morfologica sau lexicala? Aveti in lexicon si constructii pasive? (pusa de doua ori)

Intr-adevar, gramaticile normative considera diateza pasiva o categorie morfologica. Nu suntem de aceeaasi parere, motiv pentru care nu am retinut pasivul in inventarul de trasaturi si nici in lexiconul formelor.

Cum tratati cazurile de ambiguitate morfologica? (pusa de doua ori)

Voi relua explicatia data in prezentarea dictionarului. Sa presupunem ca avem cuvantul englezesc **fly**. Trasaturile de lema ne vor ajuta sa dezambiguizam mai intai partea de vorbire si astfel cuvantul **fly** va intra in dictionar cu doua leme, una pentru verb si alta pentru nume. Verbul la randul lui e ambiguu si dezambiguizarea urmatoare se va face cu trasaturile formelor flexionare. Vom avea din nou doua intrari, adnotate dupa cum urmeaza: **fly** pr12sg; **fly** 123pl.

Nu mi-e clar de ce faceti diferenta intre nume proprii si comune. (pusa de doua ori)

Avem nevoie de distinctia aceasta deoarece exista diferente de flexiune intre nume proprii si nume comune.

Adjectivul este in romana o categorie care are si ea articol? Aratati-mi, va rog, diferenta dintre un adjectiv articulat si unul nearticulat. (pusa de doua ori)

Adjectivele pot avea si articol (definit) atunci cand preceda un substantiv. Pentru combinatia **copilul frumos** (in care articolul definit sta ca de obicei la substantiv), adjectivul postnominal nu poate fi articulat. In cazul plasarii lui inaintea verbului insa, adjectivul preia articolul substantivului: **frumosul baiat**.

Nu ar fi posibil sa imbogatiti dictionarul printr-un procedeu automat, mai clar, introducand forma unui cuvant manual si apoi construind restul paradigmei cu ajutorul unui program? (pusa de doua ori)

Sigur ca ar fi posibil, dar ne trebuie tocmai un astfel de program, pe care speram sa il realizam in viitor.

Care este de fapt scopul unui asemenea dictionar? (pusa de doua ori)

Un dictionar este o resursa, iar resursele sunt multifunctionale. Un scop pe care un astfel de dictionar l-ar putea servi este acela al invatarii limbii romane de catre straini.

Poate lexiconul sa-mi indice structura interna a unui cuvant? (pusa de doua ori)

Nu poate. El poate doar sa dea o forma intreaga impreuna cu informatia relevanta pentru forma in cauza.

Din cate stiu, romana foloseste mijloace analitice pentru exprimarea gradului de comparatie la adjective. Acesta este motivul pentru care gradul de comparatie nu apare in dictionar? (pusa de doua ori)

Da, acesta este. Adjectivele cu grade de comparatie sunt socotite cuvinte alcatuite din alte cuvinte.

De ce aveti nevoie de deosebirea dintre trasaturi de lema si trasaturi de forme flexionare? (pusa de doua ori)

Distinctia in cauza a fost adoptata mai ales din considerente de descriere uniforma - dictionarul bulgar procedeaza si el astfel. Intr-o anumita masura, ea este desigur si o distinctie teoretica, numai ca noi credem ca descrierea putea sa functioneze la fel de bine si fara aceasta distinctie.

6678 de forme flexionare este un esantion prea mic de dictionar. Il veti extinde in viitor? (pusa de doua ori)

Desigur, numai ca aceasta va fi parte a unui nou proiect.

Categoria particulelor nu este convingator definita in preambulul dictionarului. Ati putea fi mai explicit in privinta motivelor care v-au determinat sa adoptati o astfel de categorie? (pusa de doua ori)

Desigur, aceasta categorie este destul de eterogena, dar n-am gasit o solutie mai buna de a lucra cu elemente care nu sunt nici adverbe nici vreo alta parte de vorbire. Astfel incat solutia noastra a fost una de extrema urgenta.

Articolul posesiv si demonstrativ sunt si ele reprezentate in dictionar? N-am putut sa le gasesc. (pusa de doua ori)

Cele doua asa-numite articole nu apar in lexicon, probabil din motivul ca nici corpusul nu le contine. Dar nu e nici o dificultate sa extindem lexiconul cu aceste categorii.

Exista ceva ce n-am inteles privitor la relatia dintre corpus si lexicon. Contine lexiconul doar formele flexionare ale cuvintelor pe care le furnizeaza corpusul? Sau contine mai mult, mai precis paradigma completa reprezentata in corpus prin, sa zicem, doua forme flexionare? (pusa de doua ori)

Daca stiti romaneste e foarte usor de verificat relatia dintre corpus si lexicon. Lexiconul e mai bogat decat corpusul. In corpus sunt cam 1500 de forme lexicale, in timp ce lexiconul cuprinde, acolo unde este cazul, intreaga paradigma careia ii apartine forma din corpus.

Acceasi trasatura este in mod alternativ inregistrata ca o trasatura de lema si respectiv de forma flexionara. De ce?

In unele cazuri, trasatura caracterizeaza doar lema (precum trasatura **gen** in raport cu numele). In alte cazuri aceeaasi trasatura este implicata in caracterizarea formei flexionare (precum trasatura **gen** in raport cu adjectivul).

Cum trateaza segmentatorul vostru lexical secventa *am dormit*? Ca pe doua cuvinte sau ca pe unul singur?

Am dormit este considerat un cuvânt compus. Toate formele verbale primesc aceasta analiza.

Cuvintele analizate morfologic exista intr-un dictionar sau sunt analizate automat?

Nu suntem siguri ca intelegem ce spuneti. Daca va referiti la cuvintele pe care le gasiti in dictionar, ele se gasesc acolo impreuna cu informatia morfologica relevanta. Dar daca va referiti la felul in care vi se livreaza informatia legata de o anumita forma lexicala pe care o cereti, bineinteles ca aceasta informatie este livrata in mod automat.

Cum este extins lexiconul, prin achizitie automata sau manual?

Lexiconul a fost extins in mod manual.

Cate cazuri sunt in limba romana?

Lasand la o parte vocativul, se disting in mod curent patru cazuri: nominativ, genitiv, dativ si acuzativ.

Am incercat sa accesez pagina cu tokenizerul si nu am gasit-o. S-a schimbat adresa initiala?

Din cate stiu, nu. Mai incearca.

Care este utilitatea analizorului morfologic?

Analizorul livreaza informatia ceruta in legatura cu o anumita forma lexicala.

Segmentatorul este independent de limba?

Segmentatorul este independent de limba in sensul ca, daca i se da un corpus de antrenament intr-o limba diferita de limba romana, el va face pentru acea limba ceea ce face acum pentru romana.

Care este utilitatea unui segmentator in prelucrarea textelor?

Segmentatorul ne ajuta sa extragem dintr-un text cuvintele mai repede si mai usor decat daca am lucra manual.

Cum analizati cuvintele compuse?

Un cuvânt compus este considerat un singur element lexical, dar desigur compus din alte elemente lexicale. Marcam cuvintele compuse prin underscore: **nici_un**.

De ce distingeți permanent între trasaturi de lema și trasaturi de forma flexionară?

În unele cazuri, trasatura caracterizează doar lema (precum trasatura **gen** în raport cu numele). În alte cazuri aceeași trasatura este implicată în caracterizarea formei flexionare (precum trasatura **gen** în raport cu adjectivul).

Aveti distincția articulat-nearticulat, dar în interiorul substantivelor articulate nu apare deosebirea hotarat-nehotarat. De ce?

Da, bună întrebare! Trebuie să incorporăm și această pereche de trasaturi, căci, evident, influențează flexiunea.



DATA RESTITUIRII

26 FEB 2024	22 APR 2024	17 JUN 2024
28 FEB 2024	24 APR 2024	23 JUN 2024
04 MAR 2024	26 APR 2024	27 JUN 2024
18 MAR 2024	06 MAY 2024	09 JUL 2024
20 MAR 2024	13 MAY 2024	12 JUL 2024
27 MAR 2024	15 MAY 2024	
28 MAR 2024	17 MAY 2024	
03 APR 2024	22 MAY 2024	
04 APR 2024	27 MAY 2024	
08 APR 2024	29 MAY 2024	
19 APR 2024	08 JUN 2024	
17 APR 2024	10 JUN 2024	

BIBLIOTECA
UNIVERSITATII



2023	22 NOV 2023	22 FEB 2024
------	-------------	-------------

